

**United States Military Academy
West Point, New York 10996**

Quantifying Situation Awareness In Janus

**DEPT. OF SYSTEMS ENGINEERING AND
OPERATIONS RESEARCH CENTER
TECHNICAL REPORT**

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

By

**MAJ E. Todd Sherrill
Operations Research Center**

**Mr. Paul West
Department of Systems Engineering**

**MAJ Michael J. Johnson
Department of Mathematical Sciences**

**Dr. Donald R. Barr
Department of Systems Engineering**

March 1998

DTIC QUALITY INSPECTED 3

19980608 122

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE MARCH 1998		3. REPORT TYPE AND DATES COVERED TECHNICAL REPORT	
4. TITLE AND SUBTITLE QUANTIFYING SITUATION AWARENESS IN JANUS				5. FUNDING NUMBERS	
6. AUTHOR(S) MAJ E. TODD SHERRILL, MR. PAUL WEST, MAJ MICHAEL J. JOHNSON, DR. DONALD R. BARR					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) USMA OPERATIONS RESEARCH CENTER WEST POINT, NEW YORK 10996				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED				12b. DISTRIBUTION CODE	
<p>13. ABSTRACT (Maximum 200 words)</p> <p>We have developed a measure of the theoretical increase in situation awareness of a tactical commander as a result of receiving data from reconnaissance, scouting and intelligence activities. We intend the measure, called "information gain" (IG), be used as a measure of effectiveness of information systems. It is based on the concept of modeling a commander's uncertainty about his adversary's disposition in terms of probability distributions over the set of states the adversary may occupy. Starting with an initial distribution, subsequent updates are calculated using Baye's formula, exploiting the operating characteristics of the sensor systems used and the search activities conducted during a sequence of time intervals. In particular, the updating process requires the probability of detection and probability of false alarm for each set of parameters involved, including the time interval, the sensors used, the area scanned by the sensors and the targets possessed by the adversary and their positions. We implemented computation of information gain for combat simulations conducted using the Janus model. This proved to be a challenge for Janus-simulated combat because we had to devise innovative methods to obtain detection probabilities for sensor-target pairs at various ranges, and also to determine the sets of cells within the battle area scanned by each sensor during each time interval. We also developed ways to account for movement of mobile targets over time. For example, a target located at a certain time may not be in the indicated location at some later time, provided the target is not killed and it is not re-detected. Thus, information about such a target may actually degrade over time. In this report we describe our approaches to these and other issues in implementing the information gain measure for JANUS applications.</p>					
14. SUBJECT TERMS JANUS				15. NUMBER OF PAGES 53	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT		

Quantifying Situation Awareness in Janus

**MAJ E. Todd Sherrill
Operations Research Center**

**Mr. Paul West
Department of Systems Engineering**

**MAJ Michael J. Johnson
Department of Mathematical Sciences**

**Professor Donald R. Barr
Department of Systems Engineering**

**U.S. Military Academy
West Point, NY 10996**

**A TECHNICAL REPORT OF THE
OPERATIONS RESEARCH CENTER
UNITED STATES MILITARY ACADEMY**

**Directed by
Lieutenant Colonel William B. Carlton.
Director, Operations Research Center**

March 1998

ABSTRACT

The United States Army intends to enhance combat power on future battlefields by skillfully exploiting information and information technology. Of the various categories of battlefield information that may ultimately contribute to combat power, information about the enemy is clearly the centerpiece. This paper reviews a newly developed method of quantifying a Blue commander's information about enemy forces, using a measure called information gain. Over an interval of time the measure represents the distance between two discrete probability distributions representing the probabilities, from Blue's perspective, that a Red vehicle is in various areas of the battlefield. When any Blue sensor scans an area of the battlefield, Blue generally gains information about the enemy disposition. The information about a detected Red vehicle may degrade over time if the vehicle is not continually observed or killed. An information degradation model is developed to account for such information reduction. We report the results of efforts to automate the information gain measure of effectiveness in Janus and briefly discuss its potential uses in combat studies.

EXECUTIVE SUMMARY

We have developed a measure of the theoretical increase in situation awareness of a tactical commander as a result of receiving data from reconnaissance, scouting and intelligence activities. We intend the measure, called "information gain" (IG), be used as a measure of effectiveness of information systems. It is based on the concept of modeling a commander's uncertainty about his adversary's disposition in terms of probability distributions over the set of states the adversary may occupy. Starting with an initial distribution, subsequent updates are calculated using Bayes' formula, exploiting the operating characteristics of the sensor systems used and the search activities conducted during a sequence of time intervals.

In particular, the updating process requires the probability of detection and probability of false alarm for each set of parameters involved, including the time interval, the sensors used, the areas scanned by the sensors and the targets possessed by the adversary and their positions. We implemented computation of information gain for combat simulations conducted using the Janus model. This proved to be a challenge for Janus-simulated combat because we had to devise innovative methods to obtain detection probabilities for sensor-target pairs at various ranges, and also to determine the sets of cells within the battle area scanned by each sensor during each time interval. We also developed ways to account for movement of mobile targets over time. For example, a target located at a certain time may not be in the indicated location at some later time, provided the target is not killed and it is not re-detected. Thus, information about such a target may actually degrade over time. In this report we describe our approaches to these and other issues in implementing the information gain measure for Janus applications.

We believe we have succeeded in making the IG measure available as an MOE for analyses in studies using the Janus model. We hope researchers, analysts and combat operations experts will find it useful. The implementation methods presented in this report can be applied to other combat simulations, so the measure is potentially available to a wide segment of the analysis community.

ABSTRACT.....	ii
EXECUTIVE SUMMARY.....	iii
TABLE OF CONTENTS.....	iv
1. INTRODUCTION.....	5
The Measure.....	6
Method of Bayesian Location Updating Applicable to Janus.....	8
Example.....	10
2. IMPLEMENTATION IN JANUS.....	14
Software Design.....	15
Janus Application Concept.....	15
Battle Space.....	16
3. DETERMINING CELLS SEARCHED.....	18
Line of Sight.....	18
View Fan.....	21
Round Earth Correction.....	23
Initial Results.....	24
4. PROBABILITY OF DETECTION.....	26
Ground Vehicle Sensors.....	29
Aerial Sensors.....	30
5. INFORMATION DEGRADATION.....	32
General Description of the Concept.....	32
Calculations.....	33
Normalizing Information Gain.....	35
Extension to Include Terrain Features.....	35
6. SUMMARY AND CONCLUSIONS.....	37
Future Work.....	38
Applying Degradation to All Searched Sectors.....	38
REFERENCES.....	39
Appendix A. Bayesian Updating.....	41
Appendix B. Software Specifications.....	43
Command Line Interface.....	43
Input File Specifications.....	43
Output File Specifications.....	44
Interactive Command Language.....	44
Errors.....	44
Logic for Update.....	45
DETEX Logic.....	46
Kills Logic.....	48
DISTRIBUTION LIST.....	50

1. INTRODUCTION

The Army has expressed heightened interest in managing information processes related to combat operations. This has generated a need in the analytic community for methods of measuring information obtained through intelligence and reconnaissance. Commonly used analytic measures of the information a commander receives are based on the flow volume or transmission rate of messages, message quality, or characteristics of the data given in the messages. In addition the cognition of, and response to, information conveyed in a given set of data depends upon the receiving commander. This human process depends on the personality, training, and experience of the commander. Attempting to measure the information gained in the receipt of data, either by looking at parameters of the raw message traffic, the message distribution system, or attempting to model a commander's cognition processes, seems difficult.

A more tractable approach appears to be to attempt to capture the amount by which a commander has been informed as a result of receiving reconnaissance and other similar data. In [1] an approach is described that involves modeling a commander's uncertainty about his enemy's disposition in terms of probability distributions. As the commander gains information about his adversary, the probability distributions are updated to reflect the new state of the commander's uncertainty. Using this approach, along with information theoretic measures related to the probability distributions [9], a measure is proposed of the changes in uncertainty brought about by the receipt of new data. This approach to modeling information gain in terms of decreased uncertainty appears to fall somewhere between approaches that model characteristics of the physical communications system and those that attempt to model human cognition and response of the decision maker.

As the discussion below reveals, calculating the measure is simple once the probability distributions are known. Developing the probability distributions, however, is much more tedious. We have used various techniques of developing these distributions in previous work [1][2][3][8][10][11]. Unfortunately, each of these techniques requires time consuming manual effort. We recognized early on that if the information gain measure were to be used by analysts and warfighters, we needed to find a way around most of the overhead involved in its computation. In order for the information gain measure to be used in analyses of simulated

combat, the measure must be automated and included in simulation post-processing tools. Since we have access to Janus we decided to automate the measure in Janus using the Janus Enhanced Tool Set (JETS). JETS is a post-processing tool developed by the Department of Systems Engineering, USMA for use in combat simulation studies. As part of the effort to automate the measure we developed an algorithm for updating the probability distributions that can be implemented in software. We chose to apply a conditional probability approach using Bayes' formula. A discussion of the underlying theory for updating probability distributions follows. We intentionally omit discussion of developing the initial (or *prior*) distribution here; it is assumed we begin with a uniform distribution of possible target locations.

The Measure

Information gain measures the Blue force's awareness of Red's disposition, over time. For our purposes, "disposition" means the number and location of Red combat systems such as tanks and armored personnel carriers. Within a time interval of duration Δt , say $(t, t+\Delta t)$, the measure is a distance measure between two probability distributions P_t and $P_{t+\Delta t}$ which we refer to as the prior and posterior distributions respectively. These distributions represent the discrete probabilities, from Blue's perspective, that a Red vehicle is in various areas of the battlefield. Consider the case of one enemy vehicle located somewhere on a battlefield partitioned into cells, so each cell has a particular probability of containing the Red system. The sum of the discrete probability values over all cells would be 1.0 with those areas of greatest likelihood having the larger values. At the beginning of the time interval $(t, t+\Delta t)$ Blue's uncertainty about the Red disposition is represented by the prior distribution P_t . If the Blue force believes that the Red vehicle is equally likely to be in any one of the cells, the prior distribution would be uniform over the cells.

When any Blue sensor scans a portion of the battlefield Blue gains information about the enemy disposition. If Blue possessed a perfect sensor, in this scan he would either determine Red's location or discover cells within the battlefield in which Red is not located. The magnitude of the new information depends on the operating characteristics of the Blue sensor as well as the outcome of its scan. For example, if a particular Blue sensor has a probability of detection (P_D) of .8 then it has a .2 probability of failing to detect an actual target's presence in a

scanned cell. Since Janus does not play false "detections," we assume the sensors possessed by Blue have zero false-alarm probability (i.e., $P_F = 0$). The cells searched by Blue during the interval $(t, t+\Delta t)$ receive updated probability assignments based on the operating characteristics of this sensor. Our method of updating the probability distribution from P_t to $P_{t+\Delta t}$ is an application of Bayes' formula [1]. The Bayesian calculations incorporate P_D and P_t values in order to update to the posterior distribution $P_{t+\Delta t}$. This posterior distribution represents Blue's new uncertainty about Red's disposition and becomes the prior distribution for the next time step, $(t+\Delta t, t+2\Delta t)$.

The prior is updated to the posterior using knowledge of which cells have been searched and the P_D of the searching sensor(s). Let $T(j)$ denote the event that there is an enemy vehicle in cell j and let $I(j)$ denote the event that Blue sensors report that there is an enemy vehicle in cell j . Suppose p_i, p_j , etc. denote individual prior probabilities of the events $T(i), T(j)$, etc. With the assumption of zero false alarm rate for Blue sensors we have, by Bayes' formula:

$$\begin{aligned} P[T(j) | I(j)] &= 1.0; \\ P[T(i) | I(j)] &= 0.0; \text{ where } i \neq j; \\ P[T(i) | \sim I(j)] &= \frac{p_i}{1 - P_D p_j}; \end{aligned} \quad (1)$$

and

$$P[T(j) | \sim I(j)] = \frac{(1 - P_D) p_j}{1 - P_D p_j}, \quad (2)$$

where " $\sim I(j)$ " indicates the event "search in cell j fails to detect the target." Equations (1) and (2) apply to the situation where Blue searches and fails to detect the target during one time interval. Equation (1) applies to a cell where Blue does not look. Equation (2) applies to a cell where a Blue sensor looks and fails to detect. Note the prior probability assigned to cell j , p_j is reduced but is not driven to zero unless the P_D of the Blue sensor is 1.0. Since the denominators in each case are identical we treat them as a multiplying constant. If Blue finds the target, the cell containing the target is assigned a cell probability of 1.0. All other cells are assigned zero

probability since Blue knows the vehicle's location. See [1] for a complete development of this formulation.

As mentioned above, information gain is a measure of the distance between the prior and posterior distributions. This distance is represented as the change in *entropy* resulting from updating the prior to the posterior distribution. Shannon defined entropy as a measure of randomness or uncertainty [9]. For our application the entropy (uncertainty) of the posterior distribution is subtracted from the entropy of the prior distribution. In this respect the information gain metric captures the decrease or increase in uncertainty concerning the location of Red systems during each time interval. This change in entropy is information gain:

$$\delta(p_t, p_{(t + \Delta t)}) = \sum p_{(t + \Delta t)} \ln(p_{(t + \Delta t)}) - \sum p_t \ln(p_t),$$

where summation is over all cells for which $p_t(p_{t+\Delta t})$ is positive [1].

Method of Bayesian Location Updating Applicable to Janus

The foregoing shows how Bayes' formula could be applied for search by a single sensor in a single cell of the battle area. Next we show how this is easily extended to searches of multiple cells in each time period, by multiple Blue sensors. We also describe the corresponding computational methods we used.

Consider first the case for a single Red target and a single Blue sensor and index the cells in the battle area by $1, 2, \dots, n$. The prior probability vector $P_t = (p_1, p_2, \dots, p_n)$ is to be updated (to the posterior vector $P_{t+\Delta t}$) at the end of each time increment, using Bayes' formula. If the target is detected and located during the time period, the posterior distribution is of the form $(0, 0, \dots, 1, \dots, 0)$, so the entropy for that target drops to zero and information gain jumps to $\ln(n)$ at the end of that time. If cells in a set $K = \{k, k+1, \dots, k+m\}$ were searched during the time period and the target was not detected, the posterior would be computed as follows.

Let $T(j)$ denote "target in cell j ," and $I(K)$ denote "target found in the set $K = \{k, k+1, \dots, k+m\}$," and let $\sim I(K)$ denote "search of cells in the set K fail to detect the target." Let P_D be the detection probability and p_j be the prior probability of the event $T(j)$, as before in Equations (1) and (2).

Case (a): posterior for cell j , $j \notin K$.

$$\begin{aligned}
 P[T(j)|\sim I(K)] &= \frac{P[\sim I(K)|T(j)] \cdot p_j}{\sum_{j \in K} P[\sim I(K)|T(j)] \cdot p_j + \sum_{j \notin K} P[\sim I(K)|T(j)] \cdot p_j} \\
 &= \frac{p_j}{\sum_{j \in K} p_j + \sum_{j \notin K} (1 - P_D) p_j} = \frac{p_j}{D}
 \end{aligned}$$

Case (b): posterior for cell j , $j \in K$.

$$\begin{aligned}
 P[T(j)|\sim I(K)] &= \frac{P[\sim I(K)|T(j)] \cdot p_j}{\sum_{j \in K} P[\sim I(K)|T(j)] \cdot p_j + \sum_{j \notin K} P[\sim I(K)|T(j)] \cdot p_j} \\
 &= \frac{p_j(1 - P_D)}{\sum_{j \in K} p_j + \sum_{j \notin K} (1 - P_D) p_j} = \frac{p_j(1 - P_D)}{D}
 \end{aligned}$$

where D denotes the common denominator in the two cases.

Computation of the posterior distribution can easily be accomplished by exploiting the fact that the denominator D is the same in both cases above. We may proceed as follows: for all j for the set K of cells searched in the time interval, replace the current prior probability the target is in cell j , p_j , by $p_j(1 - P_D)$, where P_D is the detection probability of the given Blue sensor against the target in question. Then sum the elements of the resulting vector and unitize the vector by dividing each element of the vector by the sum of the elements in the vector. This vector is the current posterior distribution at the end of the time interval in question, and it becomes the prior distribution for the beginning of the succeeding time period.

It is useful to note the posterior could also be computed by envisioning the cells in the set K were searched in some order, and the distribution was sequentially updated after each individual cell search, rather than at the end of the time period, as above. This would lead to multiplying the prior value, p_j , by the non-detection probability, $1 - P_D$, for the sequence of cells j , one at a time. The final resulting posterior, after updating with each individual cell search,

would be exactly the vector as shown above. We observe this is true, regardless of the order in which we imagine the cells in the set K are searched.

Now, consider the case (relevant for Janus computations) in which the probability of detection is a function of Blue's sensor, s , and any cell, c , in which it looks sometime during the time increment. Denote this probability by $D_{s,c}$. Moreover, let $D_{s,c} = 0$ for any cell c not "inspected" by a given sensor, s during the time interval. The probability of non-detection by all sensors looking in the j^{th} cell is the product of the probabilities of non-detection by each sensor looking in that cell in the given time period, assuming independence among the sensors. Then the posterior probability vector for the target in question, given it was not located in the time increment under consideration, is found by unitizing the vector whose i^{th} element is

$$\prod_{\text{all sensors}} (1 - D_{s,j}) \cdot p_i ,$$

using the convention mentioned above for cells not inspected by the various sensors. This posterior updating can be carried out in one operation for all entries in the prior vector (corresponding to cells making up the battle area). Thus, if \mathbf{p}_t denotes the prior vector at time t (a stochastic vector having k elements), and $\mathbf{d}_{t+\Delta t}$ denotes the non-detection probability vector for the t^{th} time interval, whose elements are composed of the values $\prod_s (1 - D_{s,j})$, then the posterior vector at time $t+\Delta t$ is $\mathbf{p}_{t+\Delta t} = \mathbf{p}_t \otimes \mathbf{d}_{t+\Delta t} / |\mathbf{p}_t \otimes \mathbf{d}_{t+\Delta t}|$ where " \otimes " denotes component-wise multiplication and " $|\cdot|$ " denotes the sum of components in the vector involved (so this division constitutes unitization of the vector $\mathbf{p}_t \otimes \mathbf{d}_{t+\Delta t}$). As mentioned before, this holds only for targets not located in the time interval; otherwise the posterior vector is of the form $(0, 0, \dots, 1, 0, \dots, 0)$, where the "1" is in the location corresponding to the cell in which the target was found.

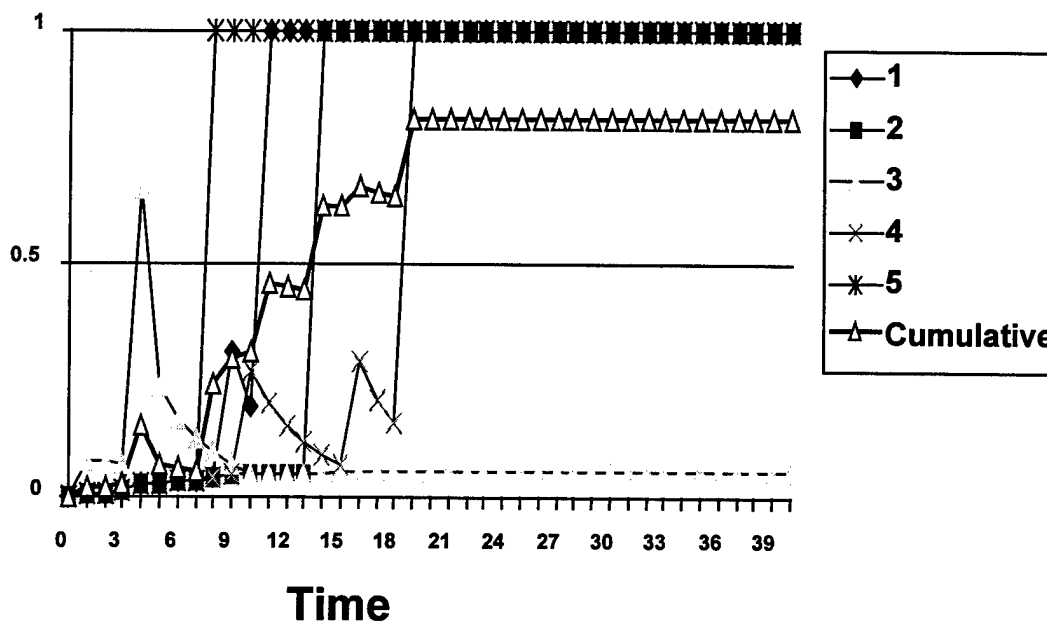
Example

Assume the Blue sensors are perfectly accurate (i.e., in each cell searched, $P_D = 1.0$ and false alarm probability is zero). If Blue detects the Red vehicle in cell j , then 1.0 is assigned to cell j and zero probability is assigned to all other cells. Blue's cumulative information gain will be at maximum value since Blue now knows all there is to know about this Red vehicle. In the case of one vehicle located in one of 100 cells, the maximum amount of information that could be attained is $\ln(100) = 4.605$ [1]. If the enemy vehicle is detected during the 1st time step, the information gain for that step would be the maximum value and the search would be over.

Likewise, the search is over when the vehicle is detected during any time step and the information gained for this time step is the maximum possible gain, 4.605, minus the cumulative gain up to the time of detection.

When Blue searches for multiple Red vehicles we simply multiply, at each time step, the information gain for one vehicle by the number of Red vehicles. In our example, assuming five enemy, the maximum gain would be $5 * \ln(100) = 23.026$. When we search a cell and find no vehicles we know that none of the five vehicles is in that cell, hence five times the gain for an individual vehicle. If we find a vehicle during the search, the information gain concerning that particular vehicle makes a jump up to $\ln(100)$ or one-fifth the total possible gain. For those vehicles remaining undetected, the gain generated by searching and not finding is now multiplied by four; we have found where four vehicles are not located. Figure 1 illustrates this approach. The graph at the right of Figure 1 represents the sum of the two plots shown in the leftmost graph.

We transform the information gain values to the scale $(-1,1)$ so that the values calculated over each Δt are relative to how much information could be known. This gives us a normalized scale and a basis for comparison.



We felt that a degradation effect was necessary to realistically model Blue's situation awareness since information is so extremely time sensitive. If a detected Red vehicle is not killed or re-detected we allow the information gain for that particular vehicle to degrade over time. Blue's spike of certainty "melts" with each passing time period as the size of the area known to contain the Red vehicle (1 cell size = 100m X 100m) expands. The rate of degradation is determined in part by the movement potential of Red vehicles. We give a detailed description of the degradation model in Section IV.

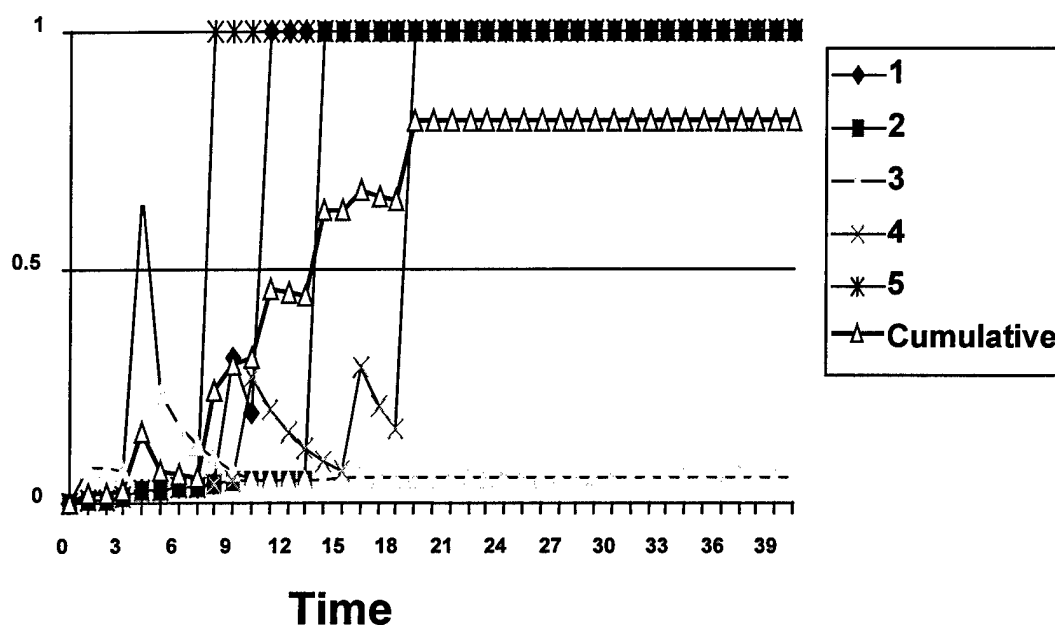


Figure 2. Hypothetical chart illustrating the individual and cumulative information gain values for each enemy vehicle over time.

A hypothetical chart of information gain concerning each enemy system is shown in Figure 2. Figure 2 also shows the cumulative information gain for Blue over all enemy systems. Computing the total information gain occurring during a time step requires a summation of the varied contributions to the total from each individual enemy system. Our software therefore must keep account of the state of each enemy system from Blue's perspective. The possible states are:

- Area - Blue is searching and finding where the vehicle is not located;
- Detection - The vehicle has been found;

- Degradation - The vehicle was detected but not killed (so it could move away from where it was detected); and
- Kill - The Blue force has killed the Red vehicle (no further movement possible).

Enemy vehicles transit from one state to another at the conclusion of a time step. The possible transitions are depicted in Figure 3.

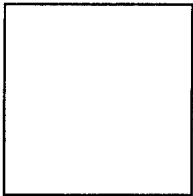


Figure 3. State transitions of enemy vehicles from Blue's perspective.

All enemy vehicles begin in the Area state. When enemy vehicles are in the Area state, Blue's information gains are determined by Blue searching and eliminating possible locations of these enemy vehicles.

Vehicles in the Detection state have been detected by at least one Blue sensor. Blue gains substantial information from a detection as can be seen in Figure 2. The spikes in information gain correspond to detections of an enemy vehicle.

Degradation begins in the time step immediately following the time step in which detection occurred provided the vehicle is not detected again or killed.

When Blue kills an enemy, Blue knows all there is to know about that enemy. We assume that this information does not decay. The dead vehicle's contribution to Blue's situation awareness reaches and remains at maximum value. This is illustrated in Figure 2 for vehicle 5 during the eighth time step, vehicle 1 during the eleventh time step, vehicle 2 during the fourteenth time step, and vehicle 4 during time step nineteen.

Note that it is possible to transit directly from the Area state to the Kill state. This may seem counterintuitive. It happens when a particular vehicle is detected and killed during the

same Δt . Note also that a vehicle can transit from Degrade to Area. This occurs when the spike in information gain due to detection has degraded over time to the point that no more is known about this particular vehicle than is known about those vehicles that have remained in the Area state.

In this regard, the information gained through searching and not finding serves as a lower bound on degradation. Vehicle 3, in Figure 2 above, degraded down to this lower bound at step 10, after being detected by Blue during the fourth time step. After step 10, vehicle 3 remained in the Area state for the rest of the battle; Blue's only awareness of vehicle 3, after step 10, was gained by finding where vehicle 3 was not located.

2. IMPLEMENTATION IN JANUS

Though the theory is simple, its implementation in the Janus model was very challenging. The Bayesian formulation presented above requires three types of data during each time stage: 1) knowledge of cells Blue sensors looked in, 2) the probability of detection (P_D) for the sensors that did the respective scanning, and 3) the prior distribution P_t . These data are not directly available in Janus runs. Nor can they be deduced from Janus output files. For example, the Janus algorithms for line of sight computations and detection of enemy vehicles are only called when two opposing vehicles are within some threshold of proximity to each other. Since information gain credits finding where the enemy is not, we need to know at each time increment what terrain cells Blue sensors have searched regardless of the presence or absence of enemy vehicles. Likewise, we need to know what the P_D would have been for each particular sensor and cell pair, had there been an enemy vehicle present when the sensor searched the cell. For our purposes, a sensor is considered to have searched a cell within a time increment if it has unobscured line of sight between its position and the particular terrain cell during that time.

Our approaches to determining which cells have been searched and the probability of detection of the searching sensor are discussed in Sections II and III; additional details concerning Bayesian updating from P_t to $P_{t+\Delta t}$ are given in Appendix A.

Software Design

Janus is the initial target for implementing this theory, but it is recognized that benefit can be obtained from its use in other models. Therefore, certain goals and constraints were imposed in the software design:

- The automated tool should require no “hooks” in the parent model. This mandates that it be either a pure postprocessing tool or a “delayed real-time” tool. The latter approach monitors the host simulation recording files and updates itself whenever the simulation dumps its event buffers to disk. For this proof of concept we adopted the former approach.
- It should be tolerant of terrain grid spacing variants. Janus terrain normally uses 100-meter resolution, but it is not required. Other models, such as ModSAF (Modular, Semi-Automated Forces), use 125-meter resolution. Our implementation converts all resolutions to 100-meter postings.
- It should be computationally inexpensive. A postprocessor with “unreasonable” computing time stands little chance of being used. Our self-imposed constraint was five minutes; actual processing time of a 45-minute battalion-on-battalion test run took about one minute.
- It should produce meaningful output for a wide variety of users. As a measure of knowledge, more should be better and scaling should be such that scenario comparisons make sense.
- It should be portable. This prototype is written in ANSI C and, although it is Janus specific, modules can be tailored to accept output from other combat simulation models.

A more complete overview of software specifications is at Appendix B.

Janus Application Concept

After loading the terrain battle space and combat system data in Janus, the following actions are implemented for each time step:

- Calculate which terrain cells all observers could be expected to scan.
- Update the p values of all terrain cells (based upon the information gained from viewed terrain cells to represent the new probabilities of the cells containing a target).
- Calculate entropy for each enemy unit based on the entire terrain cell probabilities.
- Update detections and multiply the number of aggregated sub-units by the single-unit entropy for that side.

- Account for degradation of previous detections.
- Update the entropy multiplier to reflect killed units.

Battle Space

Battle space is “determined by the maximum capabilities of friendly and enemy forces to acquire and dominate each other by fires and maneuver and in the electromagnetic spectrum.” [16]. The Operations field manual states, “Commanders use the concept of battle space to help determine how the terrain and all available combat power can be used to dominate the enemy and protect the force...” [15]. Our approach is to consider only that part of the terrain file that constitutes the Blue unit’s battle space. Doing so provides an accurate representation of information operations and minimizes unnecessary computations.

Building a battle space box is a process of enlarging a rectangle around all the entities such that it marks the maximum capabilities of friendly and enemy forces to acquire and dominate each other by fires and maneuver. This is done by reading entity locations and expanding the coordinates around them the distance of their longest-ranged sensor. If those coordinates exceed the cardinal boundaries previously set, the boundaries are pushed out to the new dimension. This process is illustrated in Figure 4.

When computed for every position of every entity for the duration of the battle, a reasonable subset of the terrain file is defined. Battle space elevation and feature data are read into memory, as are the combat system data. Feature data describes items such as buildings, bridges, dams, etc. which realistically would affect line of sight. We include the feature data for later software modifications and improvements but do not use it in this prototype. The software interfaces with the Janus files listed in Table 1. Table 1 also contains a brief description of each file’s role.

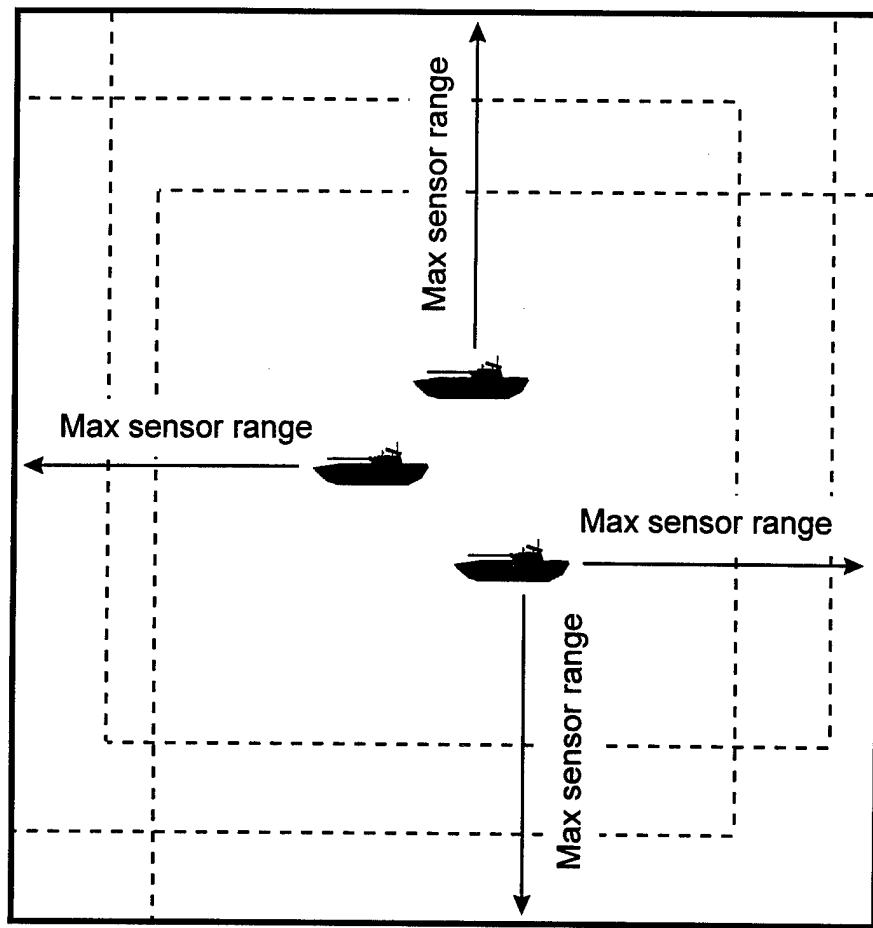


Figure 4: Battle Space

The software interfaces with the Janus files listed in Table 1. Table 1 also contains a brief description of each file's role.

File	Provides
JSCRNsssrr.DAT	Terrain file to use
TERAINttt.DAT	Elevation and feature data
DPLOYsss.DAT	Initial entity locations
FORCEsss.DAT	Scenario force structure
PPMOVEsssrr.DAT	Time, unit, location, direction of view and view fan
PPDTECsssrr.DAT	Detection events
PPKILSsssrr.DAT	Kill events
SYSTEMsssrr.DAT	System characteristics, with sensor height and range

ttt=terrain number; sss=scenario number; rr=run number

Table 1.

3. DETERMINING CELLS SEARCHED

Line of Sight

Janus is typical of Army combat simulations in the use of gridded cell terrain based on Digital Terrain Elevation Data (DTED) from the Defense Mapping Agency (DMA). This format, which covers most of the world, records the height above sea level at regular intervals. The most widely used interval, or resolution, is every 100 meters.

Elevations between these “posts” often is interpolated for greater accuracy, which is important when determining line of sight (LOS) between two specific points. Our approach, however, is to consider each 100 meter square as a horizontal cell with the elevation determined by the post at the lower left corner. We chose this convention for several reasons:

- We are concerned with potential LOS to all points within a terrain cell.

- The number of computations per cycle must permit a reasonable run speed.
- The accuracy of this method is sufficient for our purposes.

These considerations fully support the use of the Bresenham Line-of-Sight Algorithm.

The Bresenham algorithm determines the path of contiguous terrain cell elevation posts that best correspond to the observer/target (O/T) line. The algorithm finds a path from observer to target along elevation posts through iterative steps. For each iteration the algorithm evaluates the coordinates of the current posts to determine which coordinate of the post (X or Y) has the greatest error from the destination or target post. The algorithm selects the next post by moving to the nearest post in the direction of greatest error. If the coordinate errors are equal the algorithm selects the next post diagonal from the current post. Note in Figure 5 that the first post selected is 111. Post 111 is four units away from the target post in the X direction and 3 units in the Y direction. The Bresenham algorithm therefore selects the next closest post in the X direction, post 116. This technique is fast and computational inexpensive since it plots an O/T line ray in 100-meter segments and avoids floating point math by using integer arithmetic [17].

To implement this technique, we determine the X and Y positions and the height of the sensor and the target. We then proceed through the following algorithm:

1. Which is least: Δx or Δy ? (Δx is the difference between sensor X coordinate and target X coordinate). This is the initial direction of movement.
2. Plot an interim "target" point one-grid cell along the axis chosen or at the diagonal post in the direction of the target.
3. Check for LOS between the observer and the interim target. Our method is to compare the slope of the current O/T line with the last slope known to block LOS. If the current slope is greater, we assume LOS.
4. Continue to loop through steps 1 and 3 until the target is reached.

The following example illustrates this process:

Observer	
X Location	0
Y Location	0
Z (above ground)	2

Target	
X Location	4
Y Location	3
Z (above ground)	1

Table 2. Sample Observer/Target Coordinates

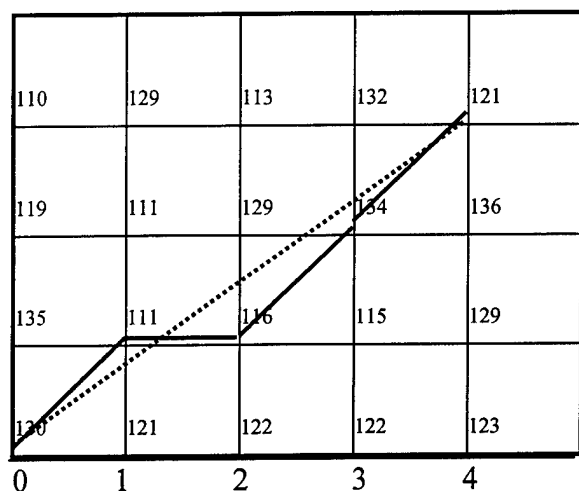


Figure 5: True vs. Bresenham Plot

The dashed diagonal line represents the true direction of sight. The solid line shows the path chosen by the Bresenham algorithm to estimate the O/T line. The numbers represent elevation postings and are marked in the lower left of their cells. We assume for our purposes that a cell whose post is visible is completely visible to our observer. Cell (2, 2, 129) is not considered in this illustration because the routine chose (3, 2, 134) as the post nearest to the O/T

line. If the two cells produce different LOS determinations, subsequent checks along the O/T line may be incorrect. This effect is mitigated through the series of near-to-far, right-to-left scans conducted for each sensor and time step (see View Fan discussion below). A cell “missed” on one near-to-far scan is likely to be picked up on the next scan, since it is not flagged as “seen.”

The observer’s elevation is the ground elevation plus the height of the sensor in the observer’s cell, in this case 132. The target is the lower left corner of each successive cell along the O/T line. We compare two slope values as we progress down the O/T line: an O/T slope and a “blockage slope.” For LOS to exist to the target, the value of its O/T slope must be greater than that of the blockage slope.

Since the observer actually may be in any part of the observer cell, we assume that cell can be seen. The initial slope therefore is $(111-132)/1 = -21$. This also becomes the initial blocking slope. The slope to cell two (2, 1, 116) is $(116-132)/2 = -8$. Since this value is greater than the blocking slope, it is considered visible. It also becomes the next blocking slope. The next step is to apply a probability that an enemy would be detected in that cell if it were there. This detection probability (P_D , discussed later) is stored in a structure representing that cell. The process repeats for the remaining cells. The slope to cell three is .67, again potentially visible and again becoming the blocking slope. Finally, the slope to the end of the O/T line, the maximum range of the system’s sensor, is -.25. This is less than the blocking slope of .67 (from 3, 2, 134), so there is no visibility to this area and P_D need not be computed.

Figure 6 shows a profile of the terrain in this example, along with the view from the observer to the progressive targets along the O/T line.

Our process must be adequately robust to assess the conditions shown in Figures 7 and 8, where the dashed lines indicate intervisibility.

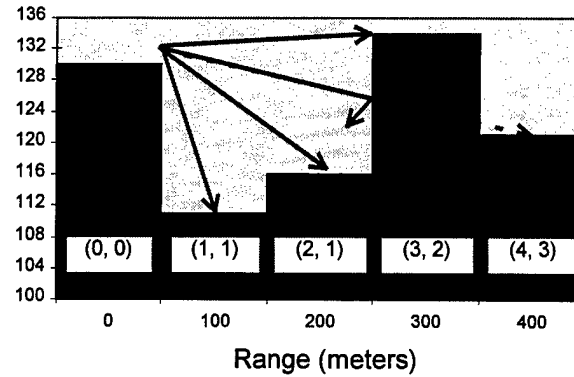


Figure 6: Terrain/View Profile

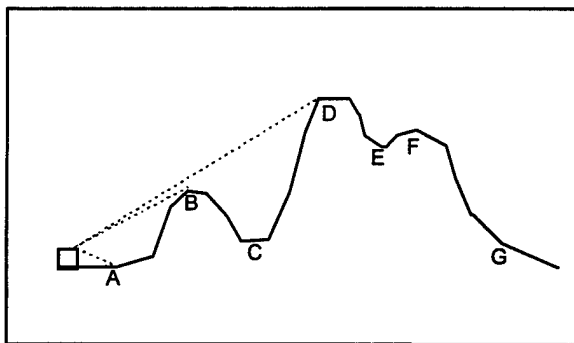


Figure 7. LOS with positive slope

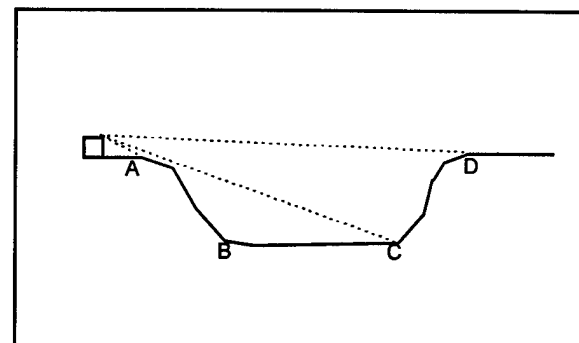


Figure 8. LOS with negative slope

View Fan

Entities in the Janus model have view fans composed of a direction of view and a right limit. This “half fan” is mirrored to provide a complete view. We use data from Janus scenario and recording files to find the X and Y coordinates of the observer and the point at the extreme sensor range along the right limit.

- $X = \text{range} \times \cos(\theta) + X_{\text{observer}}$

$$X = r * \cos(\theta) + X_{\text{observer}}$$

- $Y = \text{range} \times \sin(\theta) + Y_{\text{observer}}$

$$Y = r * \sin(\theta) + Y_{\text{observer}}$$

A near-to-far scan as described earlier checks each cell from observer to the maximum sensor range of the entity. We then loop through a series of counterclockwise moves and checks until the entire view fan has been checked. The number and size of those moves is determined by dividing the total fan angle (2 times the half-angle) by the arc length of the fan.

- Number of steps = range times (2 * fan angle); $s = r(\theta)$
- Step size = (2 * fan angle) divided by number of steps;

$$\text{step size} = \theta / \text{number of steps}$$

A temporary “seen” flag is set for each cell with LOS as the fan is checked. This flag is checked on subsequent rays so that calculations are performed on a “seen” cell only once per fan. This flag is cleared for all cells in the fan before the next one is scanned.

Three errors may be introduced by the “nearest corner” nature of the Bresenham algorithm. First, some cells near the arc may be missed, even though every cell along the arc is checked. Second, some cells may not be flagged as visible when they should be because LOS was blocked along one ray and the cell was skipped on the next. Third, some cells may appear inside or outside of the view fan.

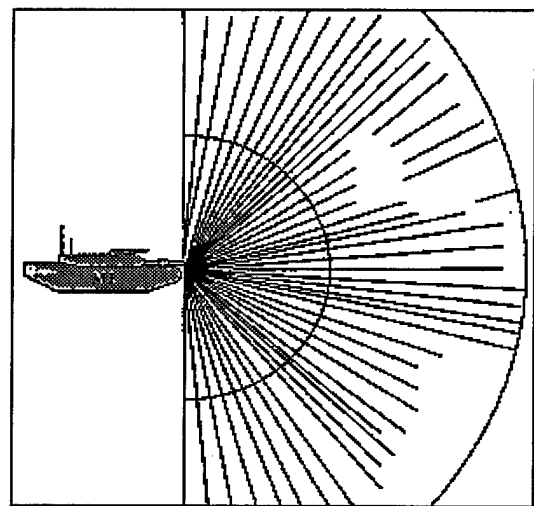


Figure 9. Janus View Fan

Consider the output shown in Figure 10 (where **S** = Seen; **B** = Blocked). Notice that cell (1, 5, 132) is not marked with an S (Seen) or B (Blocked), suggesting that it was missed. Likewise for cells (2, 4, 126), (2, 6, 127) and (8, 4, 114). Cells (6, 8, 127), (7, 7, 118) and (9, 4, 122) appear to be outside the view fan since there is no ray drawn in those cells. However, shifting the fan east (right) within the same observer’s cell initially corrects this apparent error.

Observer location (Z is ground level)	Max view (in grid cells)	View (Radians)	Half-fan (Radians)
X=5; Y=9	6	4.71 (south)	.643501

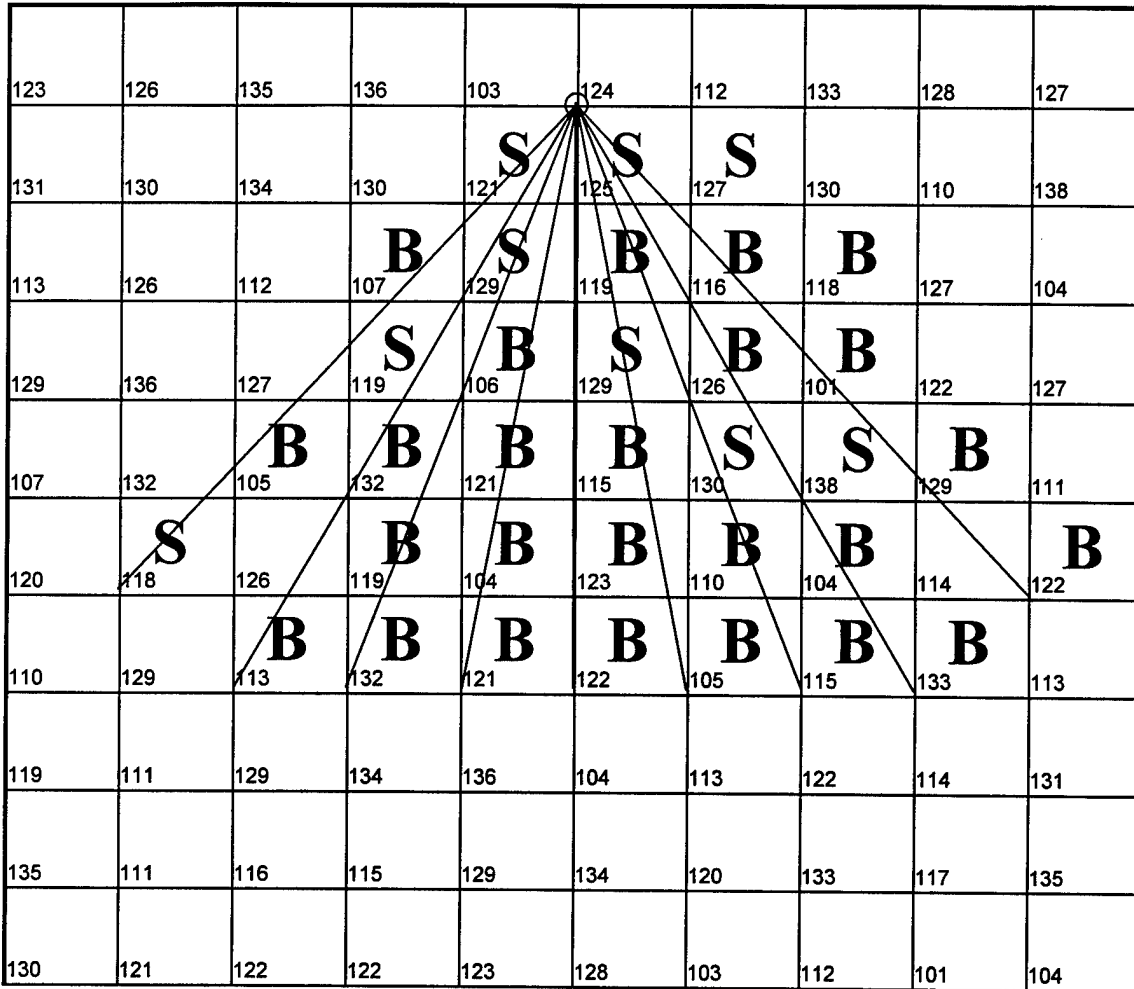


Figure 10. LOS Fan Results

Round Earth Correction

Curvature of the earth is a concern since we are dealing with sensor ranges beyond 3 kilometers. This can be corrected by using a method such as that described below [18]. This correction is not currently implemented in the prototype software.

$\Delta H = (R * \text{Csc } \theta) - R_{\text{earth}} \approx R^2 / 2 * R_{\text{earth}}$		
ΔH = elevation change	R = Range 1 cell = 100 meters	R_{earth} = Radius of Earth 6,356,766 meters

Table 3. Accounting for Curvature of the Earth

Delta H is rounded to the nearest whole number and subtracted from the elevation of the target cell. Table 4 shows how this becomes important in our considerations. A tank with a height of 3 meters which is visible at 7 kilometers in a flat-earth model actually would not be visible in the real world even on completely “flat” ground. Radars must be treated differently. In Radar Handbook, Skolnik suggests using $\frac{4}{3} \times R_{\text{earth}}$ for radar systems [19].

Range (Km)	ΔH (m)	Range (Km)	ΔH (m)	Range (Km)	ΔH (m)
1	0.078656	4	1.258502	7	3.854161
2	0.314625	5	1.966409	8	5.034006
3	0.707907	6	2.831629	9	6.371164

Table 4. Apparent Change in Elevation due to Curvature

Another factor in considering the battle space is the size of the available terrain map. Checks must be imposed to prevent a ray from exceeding the edge of the digital world.

Initial Results

Test runs were made using a U.S.-style battalion task force attack on Red defensive positions in Northeast Asia, a scenario derived from TRADOC High Resolution Scenario 31. A Plan View Display is shown in Figure 11. Blue forces, deployed in the southwest corner, advance north, and then turn east to attack into the enemy’s flank. The terrain is mountainous except in the area shown, limiting Blue’s observation. Normalized cumulative information gain traces for a typical Janus run are shown in Figures 12 and 13. Figure 12 shows increasing information gain based on cell searches, without detection. The fairly small increase can be attributed to the rugged terrain surrounding the area of operations. As actual detections and kills are considered (Figure 13), the increases become more pronounced.

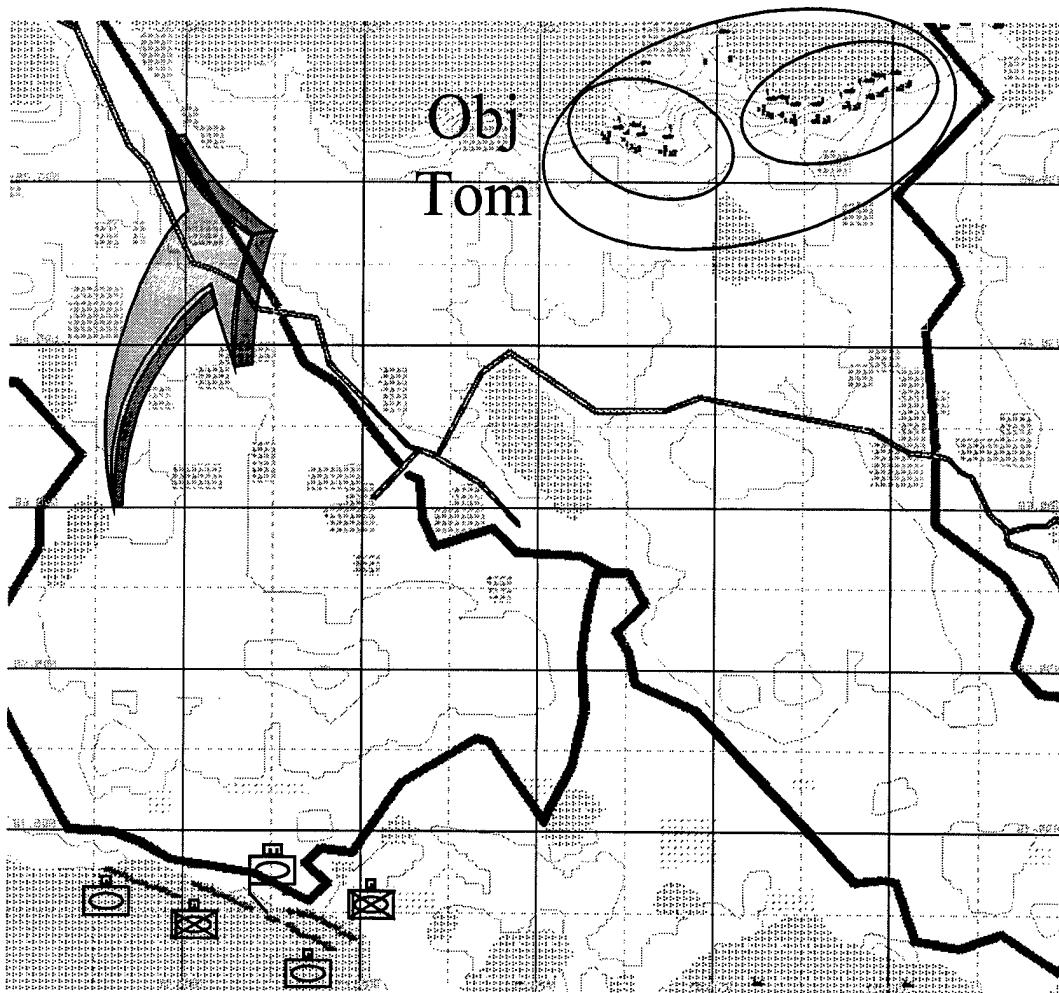


Figure 11. Scenario Plan View Display

Figure 13 shows cumulative information gain over time (solid line) and the portion attributed to search without detection (dotted line). The initial ramp in information gain, at step 2, results from detection of a BMP platoon (three vehicles). As contact is broken (at step 9), certainty decays to the value it would have been, had no detection occurred (step 10). Our expectations are confirmed by the coincidence and continuation of the line along the no-decay plot and the limiting of decay at the terrain-only plot.

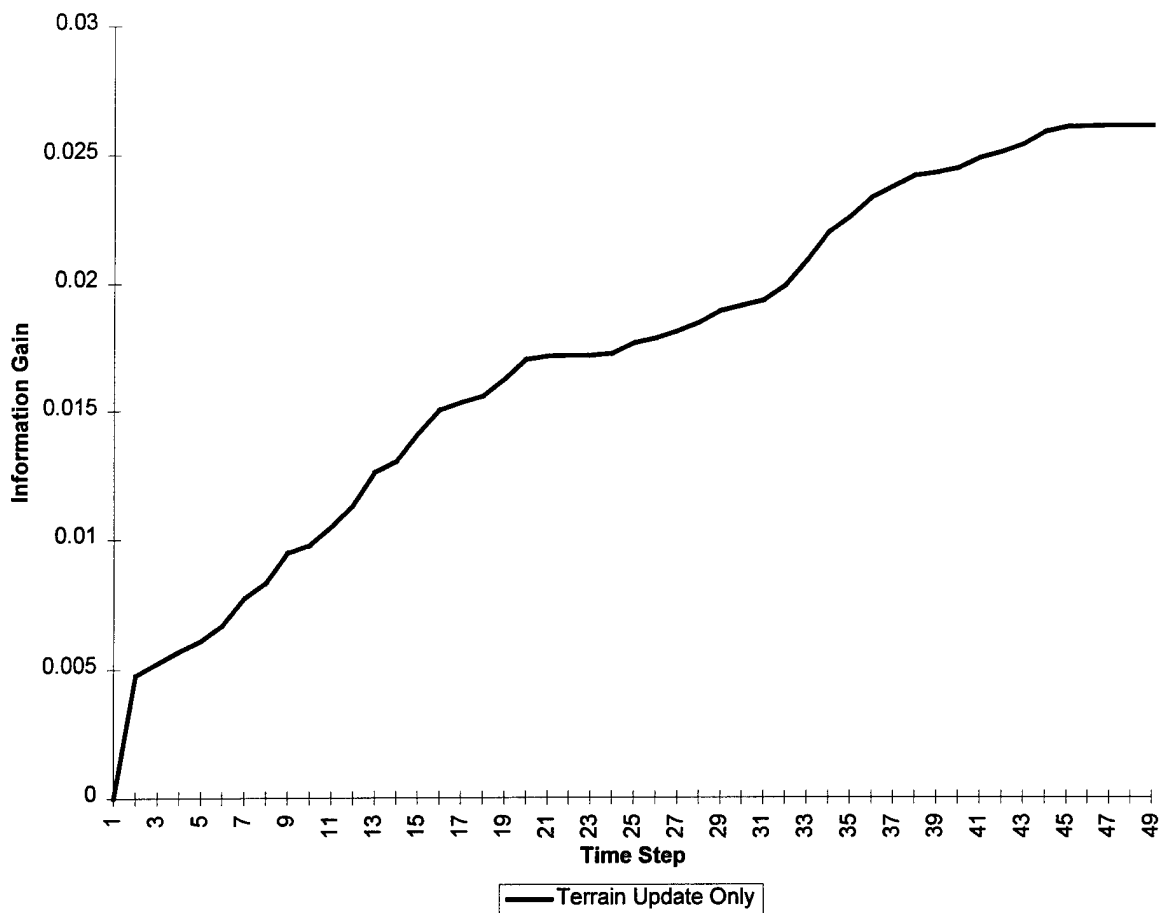


Figure 12. Gain based on terrain awareness only (Scale = 0-1)

4. PROBABILITY OF DETECTION

Independent detection modeling is required to estimate the chance that an enemy would be detected in a “seen” terrain cell if it were there. Janus computes detections only when an entity on the shooter’s target list falls within its range fan. We wish to compute P_D for each terrain cell within the observer’s range fan that the observer can see (i.e., there exists line-of-sight between the searching sensor and the terrain cell). For the current prototype, a placeholder value of $P_D = .80$ is used for all sensors and ranges.

Ideally, we prefer using the Night Vision and Electronic Sensors Directorate (NVESD) algorithm. The implementations of this algorithm are fundamentally the same in Janus, ModSAF and other models and therefore enhance portability [14]. Input data for the NVESD

model are available in the Janus scenario's SYSTEMsss.DAT file. For our purpose, P_D is the same as NVESD's P_{∞} , which is the probability that a target will be acquired in an infinite amount of time. We have been, so far, unsuccessful in our attempts to exploit this "on-board" Janus capability for our particular purposes but intend to continue the effort.

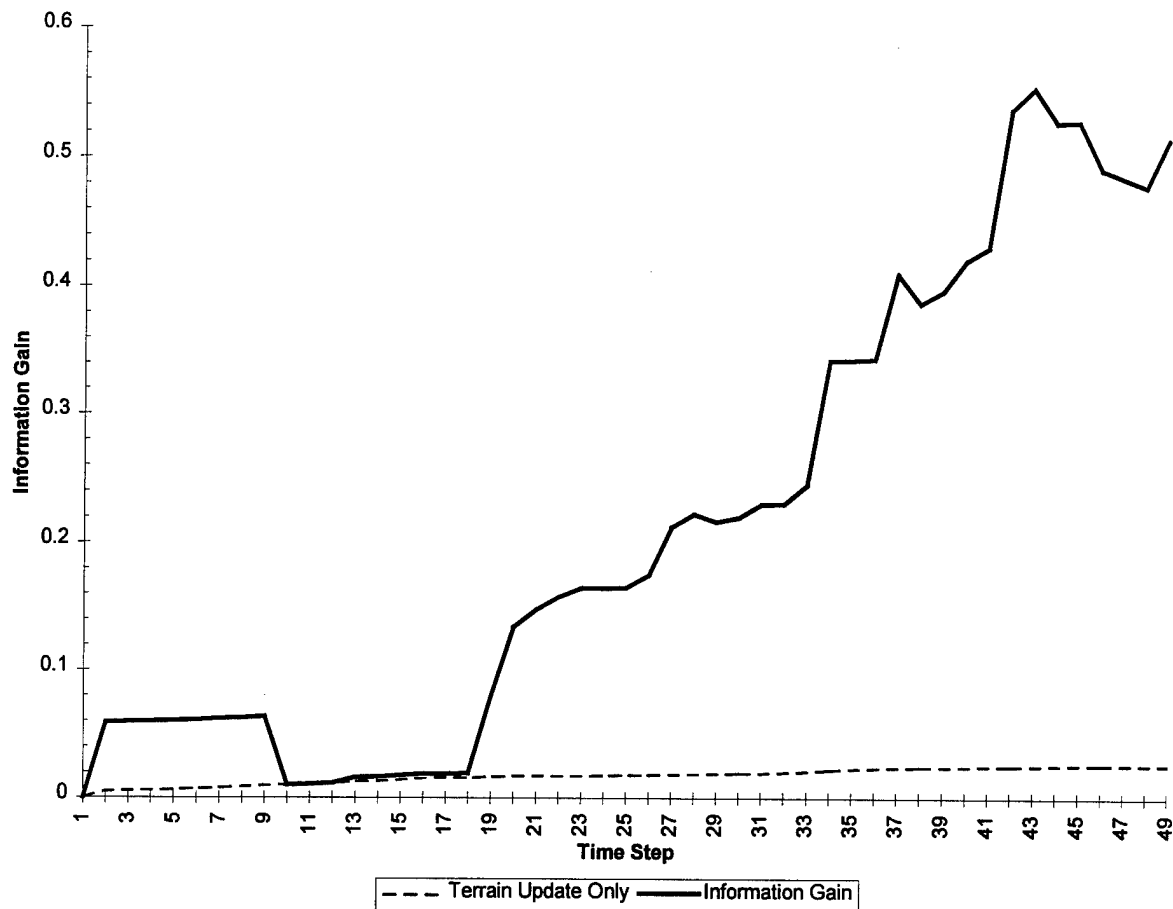


Figure 13. Comparative Entropy

An alternative approach that we have investigated is modeling detection probability using the Janus graphical validation and verification (V&V) information. In the V&V section of the Janus users interface we discovered curves that provide a representation of probability of detection data for each observer-target pair. These graphs can be defined by either primary or secondary sensors against a stationary or moving enemy target, and can be varied from simple detection to actual identification of the enemy. We decided to try to replicate the information in

these curves with a closed form function mapping the range from sensor to target into P_D . An example of the type of curves displayed in the Janus V&V section is shown in Figure 14 below.

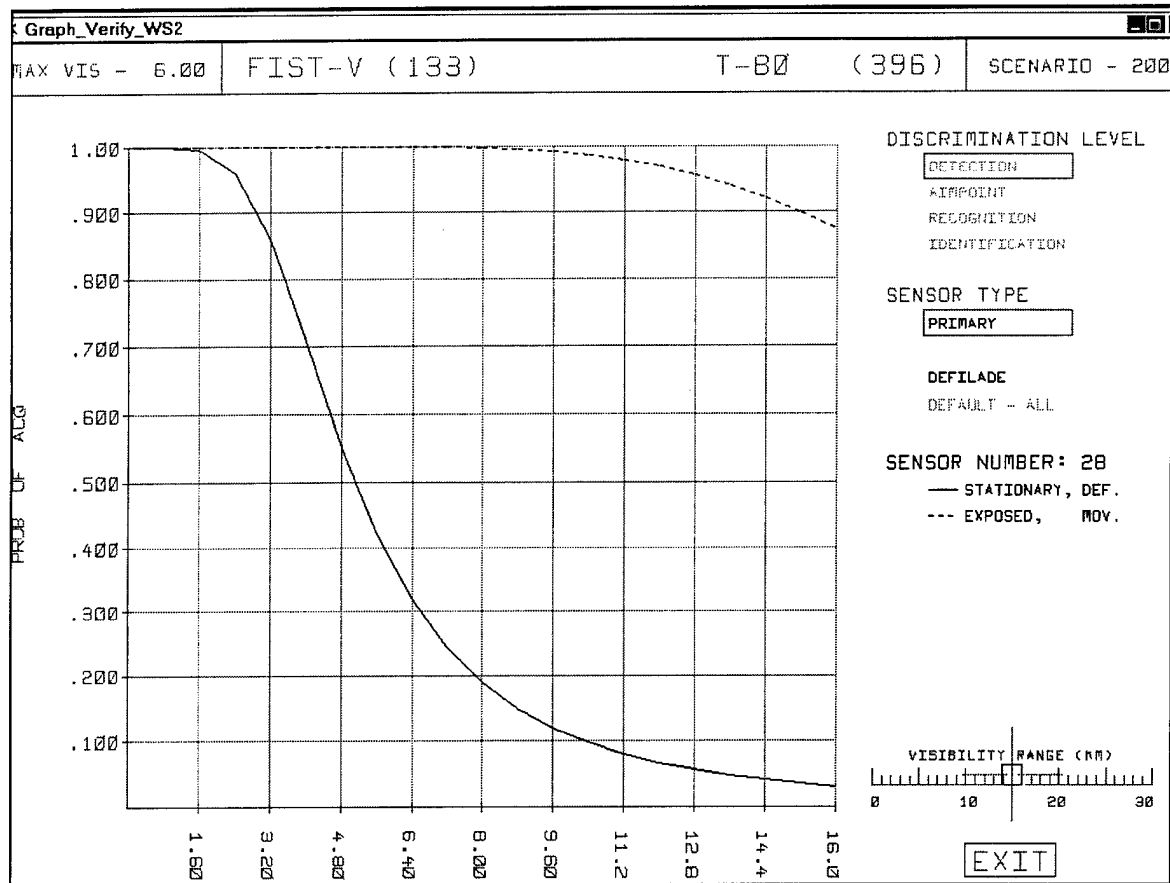


Figure 14. Probability of detection curve versus range (km) for a FistV w/ thermal sights vs. a stationary (solid line) T80.

The plot of a FISTV seeking a stationary T80 with a Thermal Sight is a continuous, monotonically decreasing function that begins at 1.0 and asymptotically approaches zero as range increases. This graph represents the probability of detection as a function of range, given the friendly vehicle/sensor type and enemy vehicle. Although this is a simplification of the Janus algorithm, it still represents the general physical nature of detection probability. As range increases, the likelihood of detection decreases.

To replicate these curves, we entered the graph for an observer-target pair, extracted ordered pairs (range, P_D) from the graph, and fit a function to the ordered pairs. We decided to

model the primary and secondary sensors on representative ground and aerial systems against an enemy T80 tank. For ground systems we chose the FistV, M1, and M2. We chose the AH-64 for aerial systems. Since the data represents probability of detection as a function of Range for each sensor, we were able to fit a curve to the data with standard mathematical techniques. The fitting procedures we attempted were polynomial curve fitting, cubic spline interpolation, fitting inverse functions to the data, conducting straight interpolation of tabled data, and performing logistics regression on the data. The method we chose was logistics regression. See [13] for a full development of these alternative approaches and the analysis supporting logistics regression.

Ground Vehicle Sensors

The 2nd order Loglog regression model for ground vehicle sensors:

$$T(y) = \ln(-\ln(1-y)), \text{ with the associated inverse transformation; } \pi(g(x)) = 1 - e^{-e^{(g(x))}},$$

Where $g(x) = \alpha + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7$, and:

- α is the y -intercept in the transformed space.
- β_1 is the coefficient multiplied by the data in the Xrange Column.
- β_2 is the coefficient multiplied by the data in the Xrange² Column.
- β_3 is the coefficient multiplied by the 1 or 0 in the FistV Column.
- β_4 is the coefficient multiplied by the 1 or 0 in the M1 Column.
- β_5 is the coefficient multiplied by the 1 or 0 in M2 Column.
- β_6 is the coefficient multiplied by the 1 or 0 in the Thermal Column.
- β_7 is the coefficient multiplied by the 1 or 0 in Optical Column.

A typical equation for $g(x)$ will look like:

$$g(x) = 16.244877 - 0.451966x_1 + 0.0066014x_1^2 - 0.313104x_3 - 0.2118893x_4 + 0x_5 \\ - 14.058842x_6 - 12.466667x_7$$

which will be coded into the detection algorithm along with $\pi(g(x)) = 1 - e^{-e^{(g(x))}}$. This is used to predict the probability of detection, P_D , for the given ground vehicle sensors against the given target. A plot of P_D versus range for this combination is shown in Figure 15.

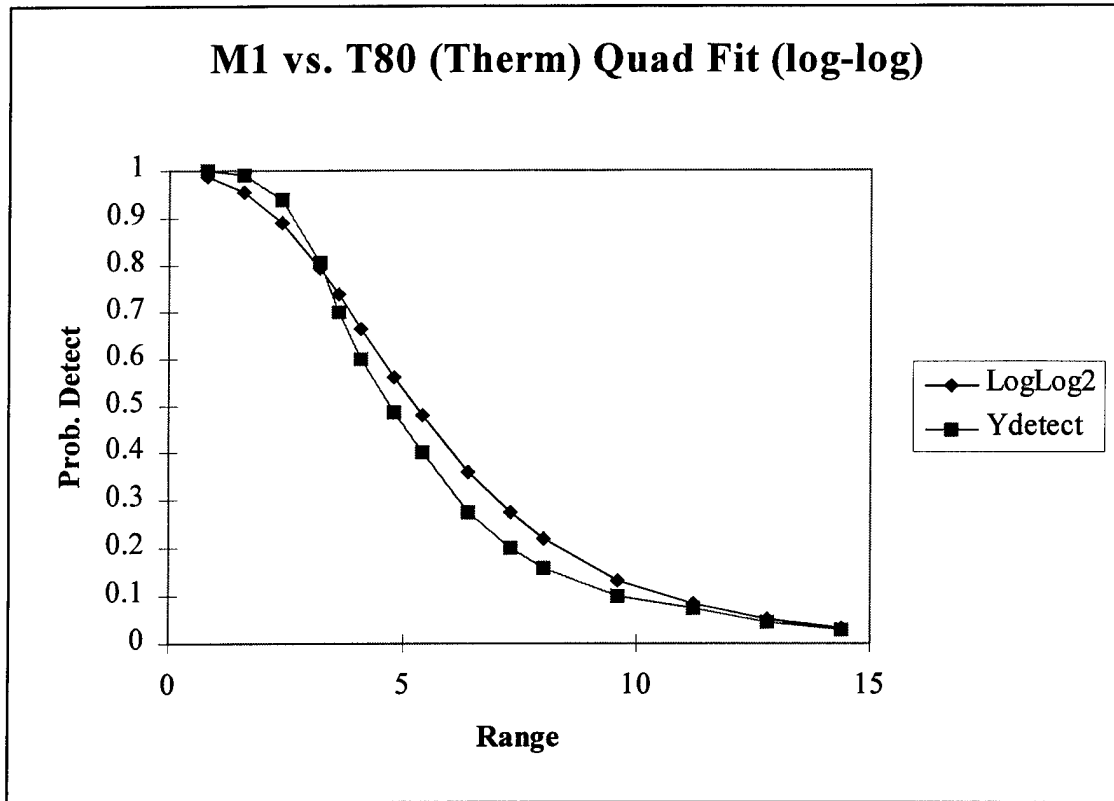


Figure15. Graphic comparison of Loglog quadratic fit to original data, (M1 (therm) vs. T80).

Aerial Sensors

The 3rd order Logit regression model for aerial sensors:

$$T(y) = \ln\left(\frac{y}{1-y}\right), \text{ with the associated inverse transformation; } \pi(g(x)) = \frac{e^{g(x)}}{1 + e^{g(x)}},$$

Where $g(x) = \alpha + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7$, and:

α is the y -intercept in the transformed space.

β_1 is the coefficient multiplied by the data in the Xrange Column.

β_2 is the coefficient multiplied by the data in the Xrange² Column.

β_3 is the coefficient multiplied by the 1 or 0 in the Xrange³ Column.

β_4 is the coefficient multiplied by the 1 or 0 in the AH-64 Column.

β_5 is the coefficient multiplied by the 1 or 0 in OH-58D Column

β_6 is the coefficient multiplied by the 1 or 0 in the Thermal Column.

β_7 is the coefficient multiplied by the 1 or 0 in Flir Column.

The final equation for $g(x)$ with estimated coefficients for these sensor-target pairs is:

$$g(x) = 866.981162 - 2.254311x_1 + 0.136956x_1^2 - 0.002840x_1^3 - 788.398867x_4 \\ - 789.333333x_5 - 67.706670x_6 - 66.928571x_7$$

This will be coded into the detection algorithm along with $\pi(g(x)) = \frac{e^{(g(x))}}{1 + e^{(g(x))}}$, which will predict the probability of detection, P_D , for aerial sensors. A plot of P_D for these sensor-target pairs is shown in Figure 16.

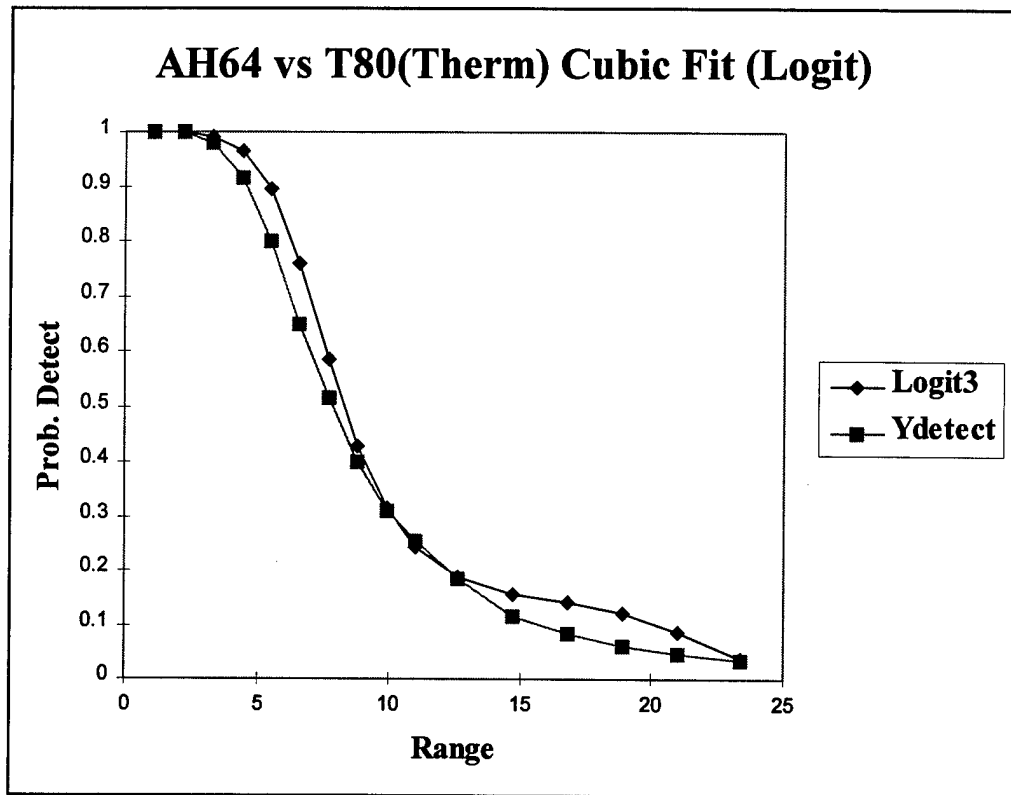


Figure 16. Graphic comparison of Logit cubic fit to original data, (AH-64 (therm) vs. T80).

The logistics regression models provide accurate models of the physical nature of detection. The chosen Loglog and Logit models provide an adequate fit to the data provided by the Janus V&V curves [13].

This alternative estimate of detection probability can be accomplished with reasonably small computation time and data storage. The model, while crude, is a marked improvement

over using the .80 constant for P_D and appears to be an adequate substitution for the NVESD model, for our application.

5. INFORMATION DEGRADATION

General Description of the Concept

The concept of degrading the information about a detected target over time if a clear line of sight with it is lost is a logical extension of the information gain model. Consider that an enemy vehicle was detected in a particular cell at a time t_0 . At time t_0 , the knowledge of that enemy vehicle is complete and a probability mass of 1 is assigned to the cell it occupies. However, as time progresses beyond t_0 , the certainty of the target's location is reduced, provided this vehicle is not detected or killed subsequent to t_0 . In this case, the possibility exists that the vehicle occupies one of the surrounding cells. In one time increment we assume it may still be located in the same cell, or it may have traveled to an adjacent cell. Assume the target is traveling at a rate r . Then at any time $t + \Delta t$ the maximum distance it could have traveled is $r\Delta t$. Ideally this would define a circle with radius $r\Delta t$ around the original location. In order to simplify the computations, to conform to the square grid pattern of cells, and most importantly, to facilitate future enhancements of the model, we assume the vehicle may radiate outward from the initial point in a square pattern. This simplification provides a slight over estimation of degradation because it assumes the vehicle could be anywhere within a square region slightly larger than the corresponding circular region.

In order for this degradation process to occur, two conditions must hold. First, for obvious reasons, the vehicle must be detected, but not killed. This would occur, for example, when the vehicle is detected beyond the maximum effective range of the weapon systems on the detecting platform. Second, the degradation will continue uninterrupted only as long as the vehicle is not detected again. Each time the vehicle is detected, the degradation process begins again when line of sight is lost.

Initially, for simplicity, we assume the target location to be distributed uniformly over all the possible cells it may occupy. This simplification is plausible because some vehicles may not move at all, others may move at a constant velocity, yet others may travel at varying velocities.

Considered over many such targets a uniform distribution may depict the average of all these scenarios. Other factors may weigh on the distribution of vehicle location, such as the terrain.

Calculations

The entropy calculations used to model the degradation of information in this scenario are similar to those in the previous sections. Tracking the uncertainty of location of a particular vehicle that has been detected can be accomplished by stepping through a time sequence. Figure 17 depicts the knowledge of the detected vehicle at the time of detection t_0 . At this time, the entropy is

$$e_0 = -\sum_{i=1}^1 p_i \ln(p_i) = \ln(1) = 0.$$

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

Figure 17. Probability distribution of vehicle location at the time of detection.

Suppose in the next time step the vehicle is not detected or killed, and the movement rate of the vehicle allows the possibility for it to occupy any of the adjacent cells. Figure 18 shows how the uniform location distribution allocates the probability to nine (3^2) possible cells. The entropy value increases to

$$e_1 = -\sum_{i=1}^9 p_i \ln(p_i) = -\ln\left(\frac{1}{9}\right) = \ln(9) = 2.1972,$$

0.00	0.00	0.00	0.00	0.00
0.00	0.11	0.11	0.11	0.00
0.00	0.11	0.11	0.11	0.00
0.00	0.11	0.11	0.11	0.00
0.00	0.00	0.00	0.00	0.00

Figure 18. Probability distribution of vehicle location after the first time step.

so the information gain over the first time step is $0 - 2.1972$.

If we assume that the vehicle can travel to one cell in each time step, the entropy after i steps is

$$e_i = \ln((2i + 1)^2) = 2 \ln(2i + 1).$$

Figure 19 shows how information gain decreases over time, assuming no re-detections in the i^{th} time step. The expression plotted is

$$G_i = e_{\max} - e_i,$$

where $e_{\max} = (\# \text{ of vehicles}) * \ln(\# \text{ of grids in the battlespace})$ is defined as the maximum uncertainty for the given scenario.

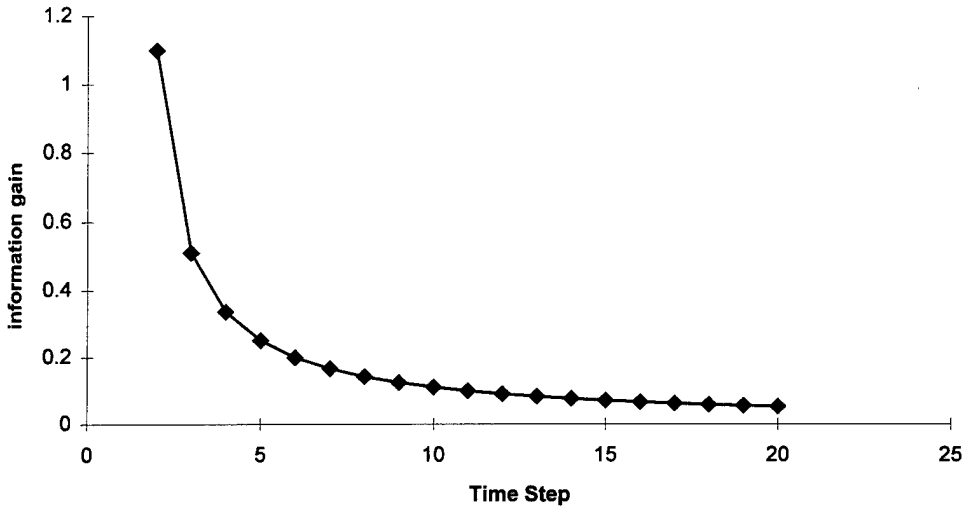


Figure 19. Cumulative information gain is a decreasing function over time, for a non-redetected mobile target.

Recall location information for a vehicle typically increases when Blue determines where the vehicle is not located, through scans with his sensors. Information gain, therefore, typically increases. When a vehicle is detected, there is a sharp increase in information gain, followed by decreases due to degradation. Since the searcher should not be penalized for achieving a

detection, the information gain does not degrade below the lower bound discussed earlier. Recall that Figure 2 demonstrates this idea using a hypothetical scenario of 5 enemy vehicles. The trajectory in Figure 2 initially followed by all five vehicles but eventually only by vehicle 3 represents the cumulative information gain for the Blue force provided there are not detections.

Normalizing Information Gain

Since information gain values are dependent upon the number of enemy vehicles and the size of the battle space (the number of cells), it may be useful to normalize the measure, as follows. Let E denote normalized information gain, so

$$E = \frac{\min \left\{ \sum_{i \in I} e_{s_i}, \sum_{i \in I^*} e_{s_i} + \sum_{j \in (I - I^*)} e_{d_j} \right\}}{e_{\max}},$$

where e_s represents the entropy, (or uncertainty) due to the search of the battlespace, e_d represents the entropy due to degradation of previously detected vehicles, I is the set of all surviving vehicles, and I^* is the set of all undetected vehicles. We note $E \in [0, 1]$. At time zero, when no cells have been searched and no vehicles have been detected, $I = I^*$ and $\sum_{i \in I} e_{s_i} = e_{\max}$, so $E = 1$. If all vehicles were detected at time t , then at that moment, $\sum_{i \in I} e_{s_i} = 0$ so $E = 0$.

Extension to Include Terrain Features

The degradation model does not consider terrain features of cells surrounding the cell where detection has occurred. Likewise, the prototype software does not use this information in its calculations of degradation. Obviously terrain attributes would influence where an enemy vehicle could travel and would help to narrow the possible location of a vehicle in time periods subsequent to detection. In principle, this could be handled simply. For instance, with the model described above, if the detected vehicle is suspected to be located in a 5×5 square of cells, and if the location distribution is assumed to be uniform, then the scenario is as shown in Figure 20. The entropy at this moment would be $\ln(25) = 3.2189$. Suppose now that the four upper right cells represent terrain that is not trafficable. The new cell probabilities can be adjusted to represent this situation. In this case each cell is assigned a mass value of $1/(25-4) = 1/21 =$

0.0476, as shown in Figure 21. The entropy value over this reduced terrain set is $\ln(21) = 3.0445$.

0.04	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.04
0.04	0.04	0.04	0.04	0.04

Figure 20. Uniform probability mass for a 5 x 5 area.

0.0476	0.0476	0.0476	0	0
0.0476	0.0476	0.0476	0	0
0.0476	0.0476	0.0476	0.0476	0.0476
0.0476	0.0476	0.0476	0.04	0.0476
0.0476	0.0476	0.0476	0.04	0.0476

Figure 21. Adjusted probability mass, accounting for terrain that is not trafficable.

There is a capability in Janus to represent the trafficability of the terrain cells. Based on terrain data within Janus, a prior distribution for target locations could be constructed. A tactical intelligence officer conducts a similar analysis when he identifies the *Go*, *Slow-go* and *No-go* terrain for the commander¹. Suppose for example, the values for a given piece of terrain are assigned as shown in Figure 22.

A prior distribution is easy to construct by normalizing these values so that the matrix represents a probability distribution. Each time the possible area containing the enemy vehicle expands due to degradation, a new prior matrix can be computed. Let $D_{i,j}$ represent the degradation matrix, (like Figure 18 above) with i rows and j columns. Similarly, let $T_{i,j}$ represent the terrain matrix, (like Figure 22 above). Then the resulting distribution would be calculated as

¹"Go, Slow-go, and no-go" are common terms used by intelligence officers and engineer officers to classify the relative trafficability of terrain.

$$P_{i,j} = \frac{(D_{i,j})(T_{i,j})}{\sum \sum (D_{i,j})(T_{i,j})},$$

where $P_{i,j}$ represents the probability of the vehicle occupying cell i,j .

Go 1.0	Go 1.0	Go 1.0	Slow-go 0.6	Slow-go 0.6
Go 1.0	Go 1.0	Go 1.0	Go 1.0	Slow-go 0.6
Go 1.0	Go 1.0	Go 1.0	Slow-go 0.6	No-go 0.01
Go 1.0	Slow-go 0.6	Slow-go 0.6	No-go 0.01	No-go 0.01
Go 1.0	Slow-go 0.6	Slow-go 0.6	Slow-go 0.6	No-go 0.01

Figure 22. Example Janus terrain matrix values.

6. SUMMARY AND CONCLUSIONS

We applied our software implementation of information gain computations in several exercises and experiments, attempting to follow rather generic scenarios. Results of these applications suggest information gain is a very useful and adaptable measure of effectiveness. It provides analysts, commanders and war planners a much needed quantitative tool related to acquisition of information by the Blue commander, as a result of data obtained from his units in the battle space.

In the course of implementing the software within the JETS post-processor, we had to overcome several obstacles related to Janus and the information gain measure. Some of the obstacles have been dealt with very crudely, and as a result, the measure implemented at the present time is itself somewhat crude. Even so, we believe it is useful in its present form. It seems clear the measure can be implemented similarly in other combat simulations.

We plan to improve this tool, to improve computation of the MOE, by

- Upgrading the mobile target model;
- Implementing improved P_D models; and
- Including effects of terrain characteristics, to facilitate automating prior distributions, and to upgrade the model of information decay with mobile targets.

Future Work

Including Vehicle Direction of Travel in the Degradation Model

An improvement to this model would account for the last known direction of travel of the vehicle. This could influence the shape of the area of the possible locations of the vehicle after it has been detected, which might be different from the square region described in Section IV. Other factors such as the disposition of the enemy forces, and enemy doctrine, could greatly effect the likelihood that the vehicle has moved, and in which direction it might possibly travel. The vehicle's actions may also depend on whether it has detected blue vehicles, and their locations.

Applying Degradation to All Searched Sectors

It makes sense that the degradation concept applied to detected mobile targets should be applied to the sectors searched by the Blue force in which no enemy vehicles were detected. This is based on the idea that even though Blue may have observed no enemy forces in a particular cell, as time continues, he is less certain that this cell remains unoccupied. Computations to implement this appear feasible. In this case the probability an enemy vehicle occupies a previously vacant cell is conditional (primarily) on the current location of the enemy vehicles. As a crude model, the computations could be based on the enemy vehicle density. The more dense the vehicles, the higher the probability one vehicle might enter a previously searched cell that Blue's sensors judges to be vacant.

REFERENCES

- [1] Barr, D., and T. Sherrill. 1996. Measuring Information Gain in Tactical Operations. Operations Research Center Technical Report, U.S. Military Academy, West Point.
- [2] Barr, D. and T. Sherrill. 1995. Estimating the Operational Value of Tactical Information. Operations Research Center Technical Report, U.S. Military Academy, West Point.
- [3] Barr, D., M. Tillman and S. Strukel. 1994. Entropy Measures of Reconnaissance. Operations Research Center Technical Report, U.S. Military Academy, West Point.
- [4] Barr, D., M. Weir and J. Hoffman. 1993. An Indicator of Combat Success. Naval Research Logistics 40, 755-768.
- [5] Janus 5.X Database Manager's Manual. 1995. Combat Simulation Laboratory, Department of Systems Engineering, U.S. Military Academy, West Point.
- [6] Khinchin, A. I. 1957. Mathematical Foundations of Information Theory. Translated by R. Silverman and M Friedman. Dover Publications, Inc., New York, 2-39.
- [7] MacWillie, S. 1992. Contributions of Reconnaissance: MOE/MOP Guide. Directorate of Combat Developments, U.S. Army Aviation Center Combat Dev. Pamphlet, Fort Rucker, AL.
- [8] Marin, J. and D. Barr. 1997. Evaluation of Intelligent Minefields. Military Operations Research (to appear).
- [9] Shannon, C. 1948. A Mathematical Theory of Communication. The Bell System Technical Journal 27, 379-423.
- [10] Sherrill, T. and D. Barr. 1996. Exploring a Relationship Between Tactical Intelligence and Battle Results. Military Operations Research 2, 17-33.
- [11] Sherrill, T., and D. Barr. 1997. Assessing Situation Awareness in Task Force XXI. Operations Research Center Technical Report, U.S. Military Academy, West Point.
- [12] Sovereign, M. 1996. Warfare in the Information Age: Vision 2010 and Changes in Joint C⁴ Doctrine. Proceedings of the 1996 Command & Control Research & Technology Symposium, National Defense University Report, D. Alberts, editor.
- [13] Sanzotta, Mickey A. and Sherrill, E. Todd "Approximating Probability of Detection in the Janus Model," A Technical Report of the Operation Research Center, U.D. Military Academy, 1997.

- [14] Courtemanche, Anthony J. and Monday, Paul "The Incorporation of Validated Combat Models into ModSAF" (ModSAF Configuration Status Accounting, <http://www.modsaf.org/doc/v&v/doc1/VV&A.html>, 1996)
- [15] Headquarters, Department of the Army, FM 100-5 Operations (Washington, Department of the Army, 1993)
- [16] Headquarters, Department of the Army, FM 100-6 Information Operations (Washington, Department of the Army, 1996)
- [17] McCoy, Hue D. "Bresenham Line-of-Sight Algorithm," Military OR Analyst's Handbook Vol 1, eds. Mark A. Youngren and Warren K. Olson, (Alexandria, The Military Operations Research Society, 1994).
- [18] Olson, Warren K. "Earth Curvature Algorithm (Sigitta)," Military OR Analyst's Handbook Vol 1, eds. Mark A. Youngren and Warren K. Olson, (Alexandria, The Military Operations Research Society, 1994).
- [19] Skolnik, Merrill I. Radar Handbook (Magraw Hill, Inc., 1970)

Appendix A. Bayesian Updating

In this implementation we use Bayes' formula to "update" the state of knowledge the Blue commander has about the presence and locations of opposing Red assets. Above we suggested Blue's "state of knowledge" could be represented by a probability distribution for each of the Red assets, and Blue's increase in information could be measured by the decrease in entropy when this distribution is changed as a result of information receipt.

Simply stated, Bayes' formula is as follows: suppose E_1, E_2, \dots, E_n is a partition of the sample space, representing "Effects" that might be observed, and let c_1, c_2, \dots, c_m be a partition of the same space into "causes." We assume the probabilities of the effects, given the causes, $P[E_i|c_j]$, are known, and the "prior" probabilities of the causes, $P[c_j]$ are also known. Bayes' formula can be used to determine the conditional probability a certain cause c_j occurred, given the effect E_i is observed, as follows:

$$P[c_j|E_i] = \frac{P[E_i|c_j] \cdot P[c_j]}{\sum_{j=1}^m (P[E_i|c_j] \cdot P[c_j])}.$$

We interpret the probability on the left to be an "updated" version of the prior, $P[c_j]$, caused by receipt of the information that E_i had occurred.

Suppose a battle area is considered to be composed of a large number of small cells C_1, C_2, \dots, C_N , and suppose reconnaissance or observation during combat can provide information implying a given cell C_j holds a given target, T , with detection probability $P_D \in (0, 1)$ (given $T \in C_j$). Similarly, suppose the false alarm rate for this recon platform on this target in this area is P_F . To simplify notation, let " $I(j)$ " denote "recon information indicates T is in C_j ," and let " $T(j)$ " denote the event " T is in cell C_j ." The current state of information, intel and recon about the location of T is represented by the current probability distribution for the location of T (which is the prior distribution for updating purposes).

Let p_j denote the prior probability of $T(j)$: $j=1, 2, \dots, N$. We use Bayes' formula to update the current distribution to take into account new information about T and, we compute the decrease in entropy of the target system to measure the value of that information.

To summarize: $P[I(i) | T(j)]$ depends on the scenario, recon tactics and capabilities of the sensors involved. We are assuming that, for the current search of cell C_j ,

$$P[T(j)] = p_j;$$

$$P[I(j) | T(j)] = P_D;$$

$$P[I(j) | T(i)] = P_F, i \neq j.$$

Then by Bayes' formula,

$$P[T(j) | I(j)] = \frac{P_D p_j}{P_D p_j + P_F (1 - p_j)},$$

and

$$P[T(i) | I(j)] = \frac{P_F p_i}{P_D p_j + P_F (1 - p_i)}; i \neq j.$$

As a special case, relevant for Janus play of combat, suppose the false alarm probability of Blue's sensor system is zero. Then application of Bayes' formula gives

$$P[T(j) | I(j)] = 1.0;$$

$$P[T(i) | I(j)] = 0.0;$$

$$P[T(i) | \sim I(j)] = \frac{P_i}{1 - P_D p_j};$$

and

$$P[T(j) | \sim I(j)] = \frac{(1 - P_D) p_j}{1 - P_D p_j}.$$

Here, " $\sim I(j)$ " indicates the event "recon in cell j fails to detect the target."

The prior is updated to the posterior using knowledge of which cells have been searched and the P_D of the searching sensor(s). Since the denominators in each case are identical we treat them as a multiplying constant. See [1] for a complete development of this formulation.

Appendix B. Software Specifications

This software calculates a measure of information gained concerning the battlefield environment in a computer combat simulation. It is based on how much of the battlefield was scanned by entities over the course of a simulation and the probability that they would have seen an enemy if one had been in their fields of view.

Data are calculated continuously for every entity and aggregated, for each side, per minute of game time. Results are stored in a table formatted for input into the Janus Evaluator's Tool Set (JETS). JETS is a postprocessing tool developed at the US Military Academy for use in analyzing data from the Janus combat simulation. Major components of this software may be adapted for use with other simulations that use grid terrain maps.

Command Line Interface

Execution requires two arguments, a three-digit scenario number and a two-digit run number, and may take one option. The command format is:

`bki sss rr [-t]`

where **bki** is the program name, **sss** is the 3-digit scenario number, **rr** is the 2-digit run number, [] denotes optional entry, - (dash) is parsed as an option flag, the **t** option turns on **test** mode, which produces a detailed test report of each module as it is being run. This slows down run speed and is suggested only for debugging and verification.

Input File Specifications

Janus version 6.x files automatically read as input are:

<code>DPLOY_{sssrr}. DAT</code>	Initial force deployment file.
<code>FORCE_{sss}. DAT</code>	Scenario force structure file.
<code>JSCRN_{sssrr}. DAT</code>	Indicates which terrain file to use.
<code>PPMOVE_{sssrr}. DAT</code>	Event recording file, with location, direction and field of view.
<code>SYSTEM_{sssrr}. DAT</code>	Characteristics file, with sensor height and range.
<code>TERAIN_{xxx}. DAT</code>	Terrain file.

Environmental variables `PRODAT`, `SCNDAT`, and `TRNDAT` are checked to determine file locations. The user is prompted for locations if those variables or the files do not exist.

Output File Specifications

Output is a comma-delimited ASCII file listing the mean entropy and BKI by side for each minute. This file is intended to be input for the JETS **.jtr* file. It therefore includes the header **TTT ENTROPY** as line 1 and the footer **%%%** as the last line. It uses the following format:

hours,minutes,seconds,entropy,bki

example: 0,0,0,10.276429,0.097310

Time Zero is a uniform value based on 1/*n* terrain cells, representing an initial baseline. The *seconds* value always will be zero and is included for conformity with the JETS format. A value for *days* may be added if analysis trends require it.

Test output produced with the **-t** option also is an ASCII file. Each group of lines contains results of logic checks generated by the *testprocs* module, which is enabled by the **-t** switch. These are the author's debugging checks and are left in to assist users in verifying the software flow. It is expected that this option rarely will be used once users become comfortable with the output.

Interactive Command Language

This program is designed to run without user interaction. Prompts will occur only under error conditions.

Errors

Error conditions will alert the user with an audible beep and a screen prompt. Where appropriate, the prompt will include a format statement and possible options. The program supports the following error messages. Italicized words represent parameters that are replaced by variable names or character strings.

1. Usage: bki (scenario #) (run #) [-t]
 Enter Scenario Number:
 Enter Run Number:
2. Path to system file is not set.
 Enter full path (eg, /users/janus/jadm/v600/projects/demo/scn)
3. Cannot open *system file*
4. Cannot open *force file*

5. Path to recording file is not set.
Enter full path (eg, /users/janus/jadm/v600/projects/demo/pps)
6. Cannot open *ppmove file*
7. Cannot open *jscreen file*
8. Path to terrain file is not set.
Enter full path (eg, /users/janus/jadm/trn)
9. Cannot open *terrain file*
10. Memory allocation for path failed.

Logic for Update

Update module flow narrative:

- I. Find **number of undetected** units by side.
 - A. Loop for all 6 sides:
 1. Loop for the number of units in each side:
 - a) If the unit's Seen flag is not set:
 - (1) Total undetected for that side is increased by the number of sub-units.
- II. Find the **entropy of detected** units by side.
 - A. Loop for all 6 sides:
 1. Loop for the number of units in each side:
 - a) If the unit's Seen flag is set:
 - (1) Sub-cells for entropy calculation is State^2 ("State" at detection is set to 1).
 - (2) Unit-entropy = $\ln(\text{sub-cells})$.
 - (3) If unit-entropy is less than the basic entropy (per unit) for that side:
 - (a) Entropy for that side is increased by the unit-entropy x number of sub units in the aggregate.
 - (b) The unit's Seen flag is increased by 2.
 - (4) Otherwise, set the Seen flag to 0 (not seen) **and** increase the number of undetected units for that side by the number of sub-units.
 - III. Find the **entropy of undetected** units by side.
 - A. Loop for all 6 sides:
 1. Add the undetected per side to a Total Undetected as it loops.
 - B. Loop for all 6 sides:
 1. If there are undetected units on that side:
 - a) Entropy for that side is set to its basic entropy (per unit) times the difference between the Total Undetected and the undetected for that side. (This considers each side versus the 5 other sides.)
 - IV. Add each side's "detected entropy" to its "undetected entropy" for a "total entropy."

DETEX Logic

I. Janus file structure, PPDTECsssrr.DAT:

Offset	Variable	Type
0	FORTRAN record mark	int
4	Game time in minutes	float
8	Observer Unit number	int
12	Observer status	int
16	Target Unit number	int
20	Target status	int
24	Range (kilometers)	float
28	Detecting sensor	char
29	Observer X coordinate	float
33	Observer Y coordinate	float
37	Target X coordinate	float
41	Target Y coordinate	float
45	FORTRAN record mark	int
49	FORTRAN record mark	int
53	2d Event Game Time	float

Detex module flow narrative:

1. Determine total number of detection events.
 - a. Find end-of-file.
 - b. Total detections = EOF / 49
2. Do while the event counter is less than total number of detection events.
 - a. Read event time.
 - b. If time = current time step:
 - (1) Skip Observer info (8 bytes)
 - (2) Read Target Unit number
 - (3) Loop for all 6 sides:
 - (a) If Target number is less than the number of units in that side:
 - (aa) If Target's Seen Flag is not set, the number of Undetected units for that side is decreased by the number of sub-units in the target.
 - (ab) Set Target's Seen Flag to 1.
 - (ac) **Break** from loop.
 - (b) Otherwise, Target Unit number is decreased by the number of units in that side.
 - c. Otherwise, skip 12 bytes (to Offset 20) to catch up to 2b sequence.
 - d. Skip 33 to align with Offset for next Event Game Time.
 - e. Increment event counter and go back to 2.

III. Detex flow chart:

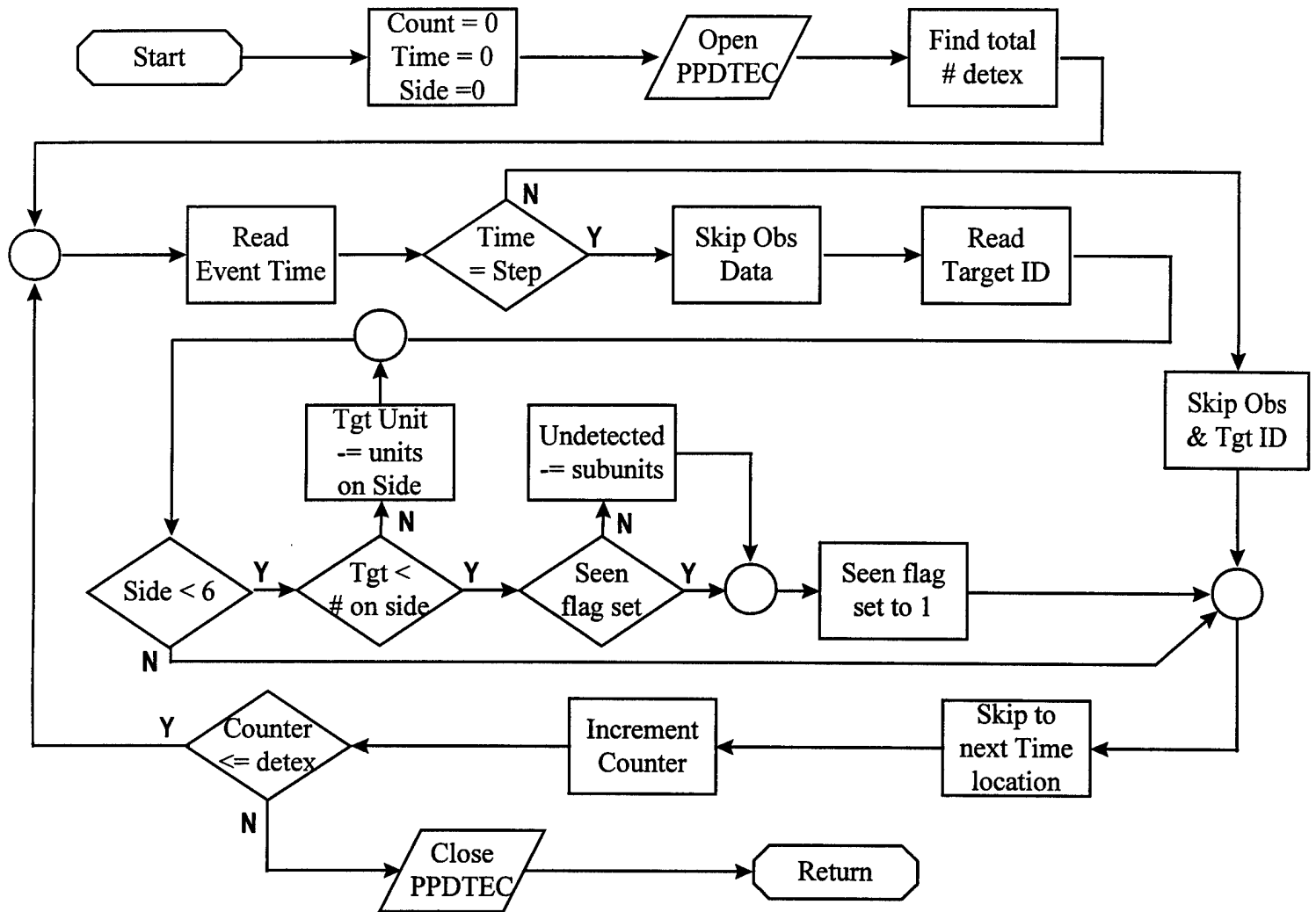


Figure 23. Detex Module Flow Chart

Kills Logic

I. Janus file structure, PPKILSsssrr.DAT:

Offset	Variable	Type
0	FORTTRAN record mark	int
4	Game time in minutes	float
8	Kill type	int
12	Victim Unit number	int
16	Victim X coordinate	float
20	Victim Y coordinate	float
24	Number of elements killed	int
28	Killer Unit number	int
32	Killer X coordinate	float
36	Killer Y coordinate	float
40	Weapon type	int
44	# Elements for this victim	char
45	Victim mounted status	short
47	Range	float
51	FORTTRAN record mark	int
55	FORTTRAN record mark	int
59	2d Event Game Time	float

Kills module flow narrative:

1. Determine total number of kill events.
 - a. Find end-of-file.
 - b. Total kills = EOF / 55.
2. Do while the event counter is less than the total number of kill events.
 - a. Read kill time.
 - b. If time = current time step:
 - (1) Skip kill type info (4 bytes)
 - (2) Read Victim Unit number
 - (3) Loop for all 6 sides:
 - (a) If Victim number is less than the number of units in that side:
 - (aa) Break from loop.
 - (b) Otherwise, Victim Unit number is decreased by the number of units in that side.
 - (4) Skip Victim X and Y coordinates (8 bytes).
 - (5) Read number of units killed.
 - (6) Decrease the victim's sub-units (aggregate) by the number of units killed.
 - (7) Skip 31 to start of next Event Game Time.
 - c. Otherwise, skip 51 to start of next Event Game Time.
 - d. Increment event counter and go back to 2.

III. Kills flow chart:

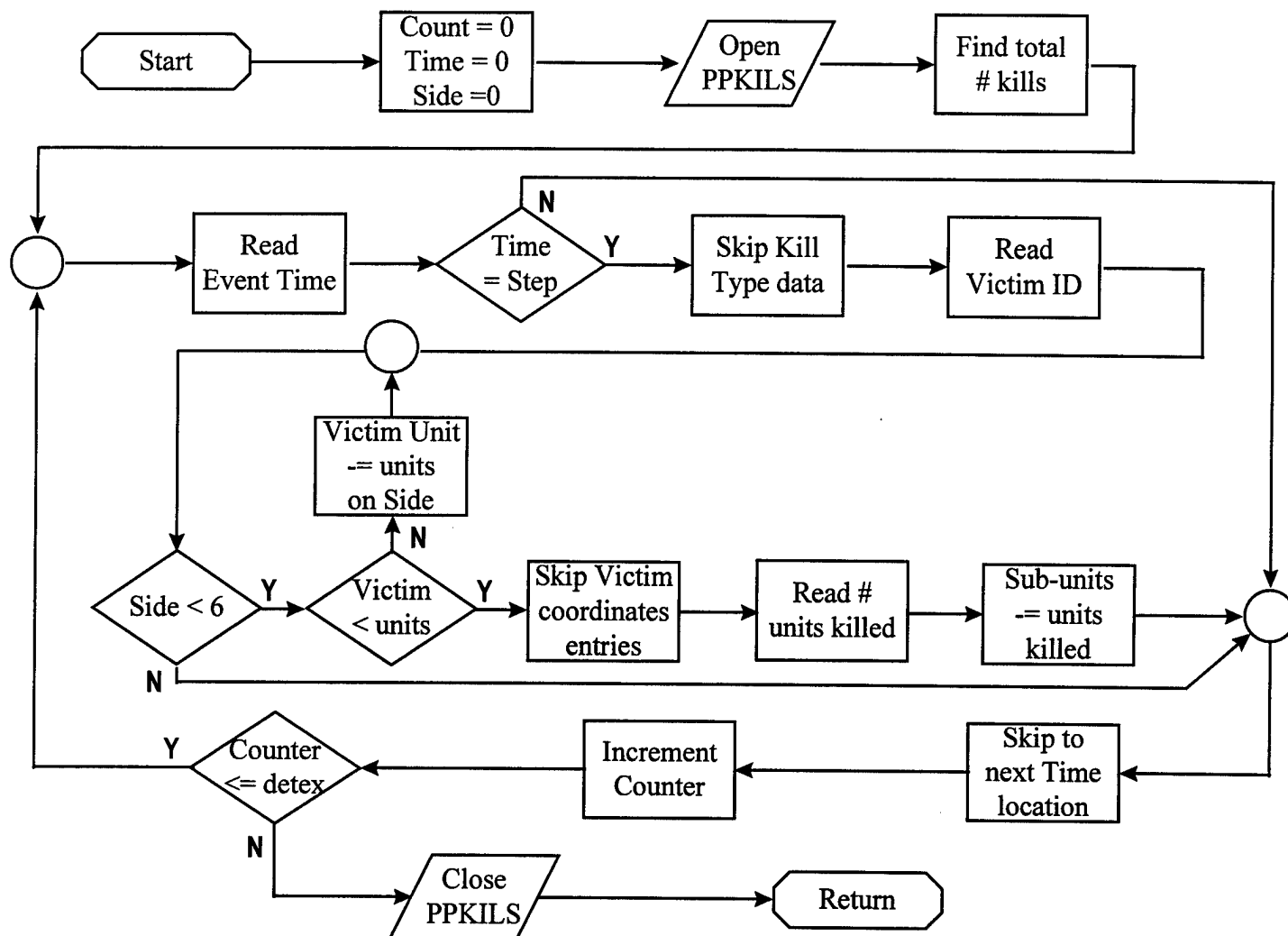


Figure 24. Kills Module Flow Chart

DISTRIBUTION LIST

<u>NAME / AGENCY</u>	<u>ADDRESS</u>	<u>COPIES</u>
Arney, COL Chris D/Mathematical Sciences	USMA Bldg 601	1
Bauman, Mr. Mike Director, TRAC	Attn: ATRC ZA 255 Sedgewick Ave. Ft. Leavenworth, KS 66027-2345	2
Bonsell, Ms. Shirley Academic Research Division	USMA Bldg 753	1
Dr. Joan Cartier Institute for Defense Analysis	1801 N. Beauregard Street Alexandria, VA 22311-1772	1
Elsaesser, Dr. Christopher The MITRE Corporation	7525 Colshire Drive McLean, VA 22102-7492	1
Fallin, Dr. Herb Research Development & Acquisition	ATTN: SARD-ZD (Dr. Fallin) 103 Army Pentagon (Rm 3E422) Washington, D.C. 20310-010	1
Fowler, Dr. Bruce W. US Missile Command Wargaming	Attn: ANSIMI-RD-AC Redstone Arsenal, AL 35898-5242	1
Holcomb, Robert Institute for Defense Analysis	1801 N. Beauregard Street Alexandria, VA 22311-1772	1
Hollis, Mr. Walter Deputy Under Secretary of the Army (Operations Research)	102 Army Pentagon Washington, DC 20301-0102	1
Huber, Dr. Reiner Univ. der Bundeswehr Munchen	D-85577 Neubiberg Germany	1
Kirby, Mr. Lee Director, USA TRADOC Analysis Command-WSMR	Attn: ATRC-W White Sands Missile Range, NM 88002	1
Lamkin, BG Fletcher Dean of the Academic Board	USMA Bldg 600	1

McCabe, Mr. Dick Chief of Concepts and Studies	Attn: ATZ-CD Ft. Rucker, AL 36362	1
Miller, Mr. Hap IDA	1801 N. Beauregard Street Alexandria, VA 22311	1
Richards, Dr. Russell The MITRE Corp.	7527 Colshire Drive McLean, VA 22102	1
Riente, Mr. John Deputy Chief of Staff for Operations & Plans	Attn: DAMO-ZD 400 Army Pentagon (Room 3A538) Washington DC 20310-0103	1
Schumante, Dr. Norman Office of the Technical Adviser	Attn: ATZK-CGT Bldg 2369 Ft. Knox, KY 40121	1
Shugart, Peter US Army TRAC-WSMR	Attn: ATRC-WB White Sands Missile Range, NM 88002-5502	1
Washburn, Prof. Alan Department of Operations Research	Naval Postgraduate School Monterey, CA 93943	1
Vandiver, Mr. E.B. III Director USA Concepts Analysis Agency	8120 Woodmont Avenue Bethesda, MD 20814-2797	1
Youngren, Prof. Mark Code OR/Ym	Naval Postgraduate School Monterey, CA 93943	1
Assistant Secretary Army (Research Development & Acquisition)	The Pentagon Washington, DC 20310	1
Commandant	US Army Armor Center Attn: ATSB-CDC Ft. Knox, KY 40121-5125	1

Commandant	US Army Aviation Center Attn: DCD Ft. Rucker, AL 36362	1
Commander USA Combined Arms Threats Dir.	Attn: CAC-T-NSC Ft. Leavenworth, KS 66027	1
Commander Naval Postgraduate School	Attn: TRAC-MTRY P.O. Box 8692 Monterey, CA 93943-0692	1
Commander US Army Research Lab	Attn: Advance Concepts & Plans White Sands Missile Range, NM 88002-5502	1
Director Program Analysis & Evaluation	ATTN: DACS-DPZ-B Washington, DC 20310-0200	1
Directorate of Combat Development	USA Field Artillery School Ft. Sill, OK 73503	1
Director Directorate of Combat Development	Attn: ATZK-CD Ft. Knox, KY40121	1
Director, USA TRDOC Analysis Command	Attn: ATRC-W White Sands Missile Range, NM 88002-5502	1
Director Directorate of Combat Development	Attn: DCD Ft. Rucker, AL 36362	1
Director Directorate of Combat Development	Attn: DCD Ft. Benning, GA 31905	1
Director Directorate of Combat Development	Attn: DCD Ft. Sill, OK 73503	1
Director Directorate of Combat Development	Attn: DCD Ft. Leonard Wood, MO 65473	1
Director TRAC-WSMR	Attn: ATRC-W White Sands Missile Range, NM 88002-5502	1

USA Armor Center SIMNET

Attn: ATSB-CD-CAD
Ft. Knox, KY 40121

1
